

# ELECTRONIC EQUIPMENT, CONTROL METHOD FOR ELECTRONIC EQUIPMENT, AND COMPUTER-READABLE STORAGE MEDIUM WITH CONTROL PROGRAM FOR ELECTRONIC EQUIPMENT STORED THEREIN

Publication number: JP2001312453

Publication date: 2001-11-09

Inventor: YAMADA KOJI

Applicant: CANON KK

Classification:

- International: G06F13/10; G06F3/12; G06F9/06; G06F9/445;  
G06F13/10; G06F3/12; G06F9/06; G06F9/445; (IPC1-7): G06F13/10; G06F9/06

- European: G06F9/445B5

Application number: JP20000128701 20000428

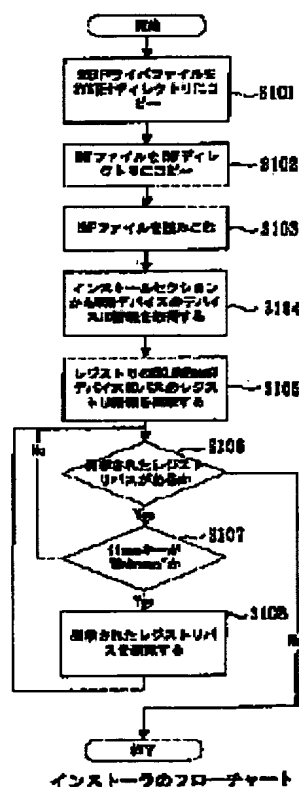
Priority number(s): JP20000128701 20000428

Report a data error here

## Abstract of JP2001312453

**PROBLEM TO BE SOLVED:** To make it easy to handle a peripheral equipment device connected to an interface where plural devices can be connected and its device driver.

**SOLUTION:** A USB driver file is copied to a SYSTEM directory (S101) and an INF file is copied to an INF directory (S102). The INF file is read in a memory (S103) and device ID information of USB in the INF file is obtained (S104). A registry of the system is read in (S105) so that it can be edited, a registry path is generated by using the device ID obtained in the step S104, and registry information below the path is listed; when a Class key is 'Unknown' (S107), information on the registry path is all deleted. When not 'Unknown', a return to a step S106 is made so as to check a next listed registry path.



(図5)

Data supplied from the esp@cenet database - Worldwide

## \* NOTICES \*

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

CLAIMS

---

## [Claim(s)]

[Claim 1] While performing the I/O to the device concerned by controlling a peripheral-device device via the device driver of the device concerned In the electronic equipment by which the registry information about registration of a device driver is registered into a registry information storage means When the device driver of a predetermined peripheral device is installed in a system, Electronic equipment which retrieves the device ID information corresponding to the device driver which should be registered with said registry information storage means, and is characterized by deleting the registry information concerned when a part of retrieved registry information is unknown.

[Claim 2] Electronic equipment according to claim 1 characterized by taking out said device ID information from an install information definition file in case the device driver of a predetermined peripheral device is installed in a system.

[Claim 3] In the electronic equipment which performs the I/O to the device concerned by connecting a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and controlling a peripheral-device device via the device driver of the device concerned The device name of the peripheral device corresponding to the port is acquired through the port which increased when it judged whether said input/output port increased and said input/output port increased. Electronic equipment characterized by performing install processing of the device driver of the peripheral device concerned when the correspondence model of specification [ the acquired device name ] is shown.

[Claim 4] Electronic equipment characterized by to report reception of this check command to a user at the peripheral-device side which transmitted the check command for a user to check the peripheral device which should be chosen as an I/O place in the electronic equipment which connects a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and performs the I/O to the device concerned via said input/output port, and received this check command.

[Claim 5] Electronic equipment which receives the status information which becomes settled according to actuation of said peripheral device or operating state from said peripheral-device device through said input/output port, and is characterized by choosing a peripheral device as an I/O place according to the received status information in the electronic equipment which connects a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and performs the I/O to the device concerned.

[Claim 6] While performing the I/O to the device concerned by controlling a peripheral-device device via the device driver of the device concerned In the control approach of electronic equipment that the registry information about registration of a device driver is registered into a registry information storage means When the device driver of a predetermined peripheral device is installed in a system, The control approach of the electronic equipment which retrieves the device ID information corresponding to the device driver which should be registered with said registry information storage means, and is characterized by deleting the registry information concerned when a part of retrieved registry information is unknown.

[Claim 7] The control approach of the electronic equipment according to claim 6 characterized by taking out said device ID information from an install information definition file in case the device driver of a predetermined peripheral device is installed in a system.

[Claim 8] In the control approach of electronic equipment of performing the I/O to the device concerned by connecting a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and controlling a peripheral-device device via the device driver of the device concerned The device name of the peripheral device corresponding to the port is acquired through the port which increased when it judged whether said input/output port increased and said input/output port increased. The control approach of the electronic equipment characterized by performing install processing of the device driver of the peripheral device concerned when the correspondence model of specification [ the acquired device name ] is shown.

[Claim 9] The control approach of the electronic equipment characterized by to report reception of this check command to a user at the peripheral-device side which transmitted a check command for a user to check the peripheral device which should be chosen as an I/O place in the control approach of electronic equipment of connecting a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and performing the I/O to the device concerned via said input/output port, and received this check command.

[Claim 10] The control approach of the electronic equipment which receives the status information which becomes settled according to actuation of said peripheral device or operating state from said peripheral-device device through said input/output port, and is characterized by to choose a peripheral device as an I/O place according to the received status information in the control approach of electronic equipment of connecting a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and performing the I/O to the device concerned.

[Claim 11] While performing the I/O to the device concerned by controlling a peripheral-device device via the device driver of the device concerned In the storage which stored the control program of the electronic equipment by which the registry information about registration of a device driver is registered into a registry information storage means and in which computer reading is possible When the device driver of a predetermined peripheral device is installed in a system, The device ID information corresponding to the device driver which should be registered with said registry information storage means. The storage which stored the control program of the electronic equipment characterized by storing the control procedure for deleting the registry information concerned when a part of retrieved registry information is unknown and in which computer reading is possible.

[Claim 12] The storage which stored the control program of the electronic equipment according to claim 11 characterized by taking out said device ID information from an install information definition file in case the device driver of a predetermined peripheral device is installed in a system and in which computer reading is possible.

[Claim 13] A peripheral-device device is connected through the input/output port which can connect two or more peripheral devices concurrently. In the storage which stored the control program of the electronic equipment which performs the I/O to the device concerned by controlling a peripheral-device device via the device driver of the device concerned and in which computer reading is possible The device name of the peripheral device corresponding to the port is acquired through the port which increased when it judged whether said input/output port increased and said input/output port increased. The storage which stored the control program of the electronic equipment characterized by storing the control procedure for performing install processing of the device driver of the peripheral device concerned when the correspondence model of specification [ the acquired device name ] is shown and in which computer reading is possible.

[Claim 14] In the storage which stored the control program of the electronic equipment which connects a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and performs the I/O to the device concerned and in which computer reading is possible A check command for a user to check the peripheral device which should be chosen as an I/O place is transmitted via said input/output port. The storage which stored the control program of the electronic equipment characterized by storing the control procedure for reporting reception of this check command to a user at the peripheral-device side which received this check command and in which computer reading is possible.

[Claim 15] In the storage which stored the control program of the electronic equipment which connects a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and performs the I/O to the device concerned and in which computer reading is possible The status information which becomes settled according to actuation of said peripheral device or operating state is received from said peripheral-device device through said input/output port. The storage which stored the control program of the electronic equipment characterized by storing the control procedure for choosing a peripheral device as an I/O place according to the received status information and in which computer reading is possible.

---

[Translation done.]

## \* NOTICES \*

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

## DETAILED DESCRIPTION

## [Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to the storage which stored the electronic equipment which connects a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, its control approach, and its control program and in which computer reading is possible.

[0002]

[Description of the Prior Art] Conventionally, peripheral devices, such as a printer, are widely used with the computer system. In the former, although the printer connected only one set of a printer to the body of a computer through the Centronics port or a serial port like RS232C in one port in many cases, the gestalt which becomes recent years and connects two or more sets of printers to one port through a more nearly general-purpose interface like the so-called USB port is increasing.

[0003]

[Problem(s) to be Solved by the Invention] Although the Windows (trade name: there are different versions, such as Windows 95, 98 and 2000, and NT) family of Microsoft Corp. is widely used as an OS of a personal computer and the support of a USB device is carried out the label side on the other hand, there are seldom many device drivers now attached as standard.

[0004] For example, the installation procedure of the USB driver in Windows 98 is manual, or a "wizard" and a called program perform install and an installer program almost similarly like the so-called SETUP.EXE of other drivers automatic thru/or by starting semi-automatically by the time of starting of OS, and PnP (plug and play) actuation of a device.

[0005] For example, if a USB device is connected to a personal computer after OS starting, OS will detect a USB device and will display the device driver install dialog of a wizard program. In this dialog, a user needs to specify correctly the directory where the device driver is stored.

[0006] If a user specifies the directory (directory on CD-ROM or a floppy (trademark) disk) where the device driver is stored through the dialog, and whether the device driver for USB devices which detected OS to the specified directory existing, and the thing which checked and corresponded exist, this will be installed and it will end.

[0007] Since the directory in which the user made a mistake was specified, when there is no device driver which does not support the detected USB device or a device driver does not exist in the directory of this \*\* primarily, OS recognizes the detected USB device as "an unknown device", and the device stops however, functioning normally henceforth. Incorrect \*\*\*\*\* data have been are registered into the setting data of OS called registry, and even if it reconnects a USB device again, in order to recognize as "an unknown device", it becomes impossible for a user to completely use a device the way things stand in this condition.

[0008] When it has been in such a condition, if a user deletes the device concerned clearly with hand control using a device manager's etc. program and does not redo right actuation, he cannot install a normal device driver. Moreover, it is difficult to carry out, if such a measure is not the user who gained a certain amount of experience, and a beginner will be puzzled. Moreover, even if the vendor which requires a large sum tariff exorbitantly is also in the user support about this kind of trifling trouble and it is a beginner, it cannot be overemphasized that to enable it to install a USB device (driver) successful certainly is desired.

[0009] Moreover, since it can connect with coincidence and two or more devices can be used for this kind of port when connecting and using devices, such as a printer, for a USB port, the new problem which did not exist is occurring conventionally.

[0010] That is, since the conventional printer was connected with the interface which can perform only the one-to-one communication link of a device with hosts, such as a centronics interface, the installer for printer drivers performed a printer and direct communication from the centronics interface, acquired information, such as a printer name, and had installed the suitable printer driver.

[0011] For example, in the case of Windows, a communication interface is expressed as a port, and the name on Windows of a centronics interface (port name) is LPT1. Then, the installer of a device (driver) opens a port called LPT1, and if the printer name which required and acquired the printer name from this port to the printer is able to judge it as the printer name to which an installer corresponds, it will perform install processing.

[0012] However, since the USB interface and IEEE1394 interface which are spreading in recent years can connect two or more devices to one interface and the address is moreover assigned dynamically, a port name cannot be decided beforehand. Therefore, by the conventional method of investigating the port name decided beforehand, the suitable printer driver by the installer could not be installed, but the burden was to be forced upon a user.

[0013] It is as follows when a trouble is mentioned more concretely. The USB interface in Windows of Microsoft is not recognized to be a port unless a device is connected with a host. And it cannot predict beforehand what kind of name even if a USB interface can connect a maximum of 127 sets of devices and is recognized to be a port, it becomes. Therefore, it became difficult to detect a connection device (model), for example, a printer, using the port name beforehand decided like before, and there was a problem that a user had to choose a suitable driver.

[0014] Or although a printer driver is also automatically installable by using the plug-and-play function in Windows, since application original with a vendor etc. is not installable, an original function cannot be added. Therefore, an installer is indispensable to install of a printer driver.

[0015] The above problem is applied similarly in the device connected through networks, such as Ethernet (trademark), for the purpose of it not being special to the printer of USB port connection, and being able to predict neither a device name nor the address beforehand.

[0016] Moreover, in the configuration to which an external instrument is connected through networks, such as Ethernet, it is not rare that two or more devices (for example, printer) of the same function are connected concurrently to the port and network, either, for example, two or more printers of the same model may be connected to the port and network.

[0017] The method of distinguishing a printer by attaching an identifier like nickname apart from a product name was common. However, this approach does not understand which printer anymore in the printer which wants to print truly, when you had forgotten nickname. Therefore, it had to check from which printer it would print to a trial and it would come out. Then, a method of distinguishing the printer which wants to print is desired, without printing by trying.

[0018] The technical problem of this invention solves the above-mentioned problem, anyone can install in a system rightly the device driver connected to the interface which can connect two or more set machines, such as a USB port, and it can identify easily external instruments, such as a printer connected to the interface which can connect two or more set machines, such as a USB port, and is to enable it to check easily devices, such as a printer which he wants to use.

[0019]

[Means for Solving the Problem] In order to solve the above-mentioned technical problem, it sets to this invention. While performing the I/O to the device concerned by controlling a peripheral-device device via the device driver of the device concerned In the storage which stored the electronic equipment by which the registry information about registration of a device driver is registered into a registry information storage means, its control approach, and its control program and in which computer reading is possible When the device driver of a predetermined peripheral device is installed in a system, The device ID information corresponding to the device driver which should be registered was retrieved with said registry information storage means, and when a part of retrieved registry information was unknown, the configuration which deletes the registry information concerned was adopted.

[0020] Or when installing the device driver of a further predetermined peripheral device in a system, said device ID information adopted the configuration taken out from an install information definition file.

[0021] Or a peripheral-device device is connected through the input/output port which can connect two or more peripheral devices concurrently. In the storage which stored the electronic equipment which performs the I/O to the device concerned by controlling a peripheral-device device via the device driver of the device concerned, its control approach, and its control program and in which computer reading is possible The device name of the peripheral device corresponding to the port is acquired through the port which increased when it judged whether said input/output port increased and said input/output port increased. When the correspondence model of specification [ the acquired device name ] was shown, the configuration which performs install processing of the device driver of the peripheral device concerned was adopted.

[0022] Or the electronic equipment which connects a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and performs the I/O to the device concerned. In the storage which stored the control approach and its control program and in which computer reading is possible The check command for a user to check the peripheral device which should be chosen as an I/O place was transmitted via said input/output port, and the configuration which reports reception of this check command to a user at the peripheral-device side which received this check command was adopted.

[0023] Or the configuration which receives the status information which becomes settled according to actuation of said peripheral device or operating state from said peripheral-device device through said input/output port, and chooses a peripheral device as an I/O place according to the received status information adopted in the storage which stored the electronic equipment which connects a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and performs the I/O to the device concerned, its control approach, and its control program and in which computer reading is possible.

[0024]

[Embodiment of the Invention] Hereafter, the gestalt of operation of this invention is explained with reference to drawing.

[0025] With a [operation gestalt of \*\* 1st] book operation gestalt, the processing which installs the device driver on Windows (trade name) is illustrated. Anyone enables it to install a device driver in a system rightly with this operation gestalt, although a setup is performed with reference to the so-called INF in the case of install of a device driver, without needing explicit deletion actuation etc. by deleting the registry pass which corresponds from registry automatically, when the device which searches the registry of a system and corresponds is an "unknown" device.

[0026] The hardware configuration for realizing this operation gestalt is explained to drawing 1 . The configuration of illustration is very common as a configuration of a well-known personal computer.

[0027] In drawing 1 , the control section (CPU is called henceforth) in which a sign H01 manages control of this operation gestalt, the keyboard into which H02 inputs various data, and H03 are the color display machines which display the image data currently stored in RAMH04, and it consists of CRT or a LCD display.

[0028] The read only memory (it is called Following ROM) which memorizes beforehand the control procedure with which a sign H04 controls the whole equipment, and the required information on other, the random access memory (it is called Following RAM) for which H05 is used as a work area, and H06 are hard disks with which R/W and program of various data are stored. the below-mentioned setup program is supplied in the condition of having been stored in the hard disk H06, non-illustrated CD-ROM, or the floppy disk, and is performed on RAMH05 (or the virtual-memory device further mounted on the hard disk H06 -- using).

[0029] A sign H07 is a USB interface and the USB device of H08 is connected. As a USB device H08, devices of arbitration, such as a printer, a keyboard, a mouse, a tablet, a sound device, and a digital camera, can be considered. It is a data bus, H09 consists of an address bus, a data bus, various control signal lines, etc., and it is used in order to transmit various data.

[0030] The file organization of the device driver installed in drawing 2 with this operation gestalt is shown. USBDEV.SYS of drawing 2 is the file of a USB device (printer in this case) driver, and is described by the gestalt of a kind of execution file dynamically incorporable into a system.

[0031] Moreover, USBDEV.INF is a configuration file used when installing the above-mentioned driver. From the extension, it is called the common-name INF file. SETUP.EXE is the install program which carried out this invention. Each file of drawing 2 is supplied in the condition of having been stored in the hard disk H06, non-illustrated CD-ROM, or the floppy disk. For example, when supplying the printer of USB connection, it shall store in CD-ROM or a floppy disk, and these files shall be supplied with the printer.

[0032] In addition, the driver of the USB port itself shall be installed separately.

[0033] The contents of the above-mentioned USBDEV.INF are shown in drawing 3 . It is the text file which arranged the value which an INF file should define a section by the identifier surrounded by [] like illustration, and should be defined as the mnemonic and parameter of a predetermined control parameter after that, and among drawing, a left end figure is a line number and does not make some data.

[0034] The device of the [Version] section of the 4th line is a USB device from the 1st line, and it is shown that the vendor (Provider) of a device is "Qanon."

[0035] As for the continuing section [DestinationDir], (the 5th line to the 6th line) shows that an install place directory is a WINDOWS\SYSTEM directory.

[0036] The [Manufacturer] section (- of 7th line the 8th line) defines above Provider and the relation of following Qanon.Section.

[0037] [Qanon.Section] (from the 9th line to the 10th line) defines install processing of the device ID "USB\VID\_04 A9&PID\_1052"

of USB. That is, it defines performing install of a device ID "USB\VID\_04 A9&PID\_1052" according to the approach of the below-mentioned [Qanon.USB] section. Moreover, it is defined here that a name is "Qanon USB Device."

[0038] In a [Qanon.USB] section (from the 11th line to the 13th line), it defines that the file copy at the time of install follows the information on the below-mentioned [Qanon.CopyList] section. Moreover, here shows that the addition of registry information follows the information on the below-mentioned [Qanon.AddReg] section.

[0039] It is shown that a [Qanon.CopyList] section (from the 14th line to the 15th line) installs a usbdev.sys file ( [drawing 2](#) ). A [Qanon.AddReg] section (from the 16th line to the 23rd line) is the information which should be added to the registry of a system. Here, the name of a driver, the classification of a device, a device ID, etc. are defined.

[0040] Next, with reference to [drawing 4](#) , it explains per processing of the registry at the time of becoming "an unknown device" at the time of install. [Drawing 4](#) shows the condition of having opened the entry of the device concerned using the registry editor. The registry information on USB TEBAISU is recorded below on HKEY\_LOCAL\_MACHINE\Enum\USB. Although registry is recorded on the hard disk H06 in the form of the data file in fact, if it accesses using a registry editor, it will be displayed as a layered structure like illustration. " \" shows a hierarchy's boundary like the delimiter of the directory in a file system.

[0041] If install actuation is performed in this TEBAISU, data will be recorded into the serial number key which a device has in less than [ HKEY\_LOCAL\_MACHINE\Enum\USB\VID\_04 A9&PID\_1052 ] like [drawing 4](#) .

[0042] [Drawing 4](#) shows the device of a serial number 000000, and the Class information on this registry information expresses the class of device. When an installer is normally performed using an INF file like [drawing 3](#) , if it is a USB device, this Class information will serve as "USB", but since the directory in which the user made a mistake was specified on the occasion of install, when there is no device driver which does not support the detected USB device or a device driver does not exist in the directory of this \*\* primarily, the value of this Class information serves as "Unknown" which shows "an unknown device" like [drawing 4](#) .

[0043] Next, the flow chart of the install program of this operation gestalt is shown in [drawing 5](#) . The program of illustration is mounted as a part of SETUP.EXE of [drawing 2](#) , is stored in CD-ROM or a floppy disk, and is supplied.

[0044] At step S101 of [drawing 5](#) , USBDEV.SYS which is a USB driver file is copied to a WINDOWS\SYSTEM directory. Then, at step S102, USBDEV.INF which is an INF file is copied to a WINDOWS\INF directory.

[0045] At step S103, a USBDEV.INF file is read into memory, in step S104, [Qanon.Section] in an INF file is looked for, and the device ID information on USB is acquired. Here, the second parameter of the right-hand side of the 10th line of the INF file of [drawing 3](#) serves as a device ID of USB.

[0046] It reads so that the registry of a system can be edited, and HKEY\_LOCAL\_MACHINE\Enum\ is added before the device ID acquired at step S104, registry pass is made from step S105, and the registry information below this pass is enumerated. If it is the contents of registry of [drawing 4](#) , the devices of the serial number 000000 equivalent to a device ID "USB\VID\_04 A9&PID\_1052" will be enumerated.

[0047] In the loop formation of steps S106-S108, if two or more devices are registered into the registry pass of the same device ID, sequential listing of these will be carried out.

[0048] At step S106, it judges whether the enumerated registry pass remains. Install processing is ended, if it remains and does not progress and remain in step S107. At step S107, it judges whether the Class key is "Unknown." If it is "Unknown", it progresses to step S108, and if it is not "Unknown", in order to investigate the registry pass with which degrees were enumerated, it returns to step S106. At step S108, all the information on registry pass that the Class key is "Unknown" is deleted.

[0049] Thus, since the directory in which the user made a mistake was specified when this program was install of the target device driver, There is a device driver which does not support the detected USB device, or Since the value of this Class information can delete all the information on the registry pass used as "Unknown" which shows "an unknown device" like [drawing 4](#) when a device driver does not exist in the directory of this \*\*, first of all, henceforth, It operates reconnecting a USB device etc., and if a right directory is specified, install of the driver of USB devices, such as a printer, can be then performed normally.

[0050] Therefore, a user can install a USB device by himself easily, without not needing actuation in which an explicit driver is deleted using a registry editor etc. like before, and needing large sum user support.

[0051] It cannot be overemphasized that it can store and provide for the install media of arbitration without limiting SETUP.EXE which carried out this invention to a floppy disk or CD-ROM.

[0052] In addition, although USB\VID\_04 A9&PID\_1052 were used as a device ID of a USB device with the above-mentioned operation gestalt, it cannot be overemphasized that the other device ID is also applicable. Moreover, although the USB port was illustrated above, even if the above-mentioned technique is the interface which connects two or more external instruments, for example, an interface like IEEE1394, it cannot be overemphasized that it can carry out similarly.

[0053] Although lessons was taken from actuation of the registry at the time of installer activation and it was shown above [the 2nd operation gestalt], below, it explains to a detail per configuration of the installer suitable for the device dynamically connected to ports, such as USB.

[0054] It is an installer corresponding to the USB printer which operates on Windows of Microsoft also with this operation gestalt. In the case of a USB printer, two device drivers are needed. One is a printer driver and another is the USB driver of lower order more. A USB driver is a device driver for transmitting and receiving data by USB. Therefore, it is necessary to install two device drivers in the installer corresponding to a USB printer.

[0055] The installation procedure in this operation gestalt serves as the following. a user -- the installer of this operation gestalt -- starting (or auto-boot being carried out by a wizard's etc. program) -- since the screen of [drawing 7](#) is displayed, a user connects a USB printer with a host. Thereby, a USB driver is installed through the below-mentioned installer program.

[0056] Install processing of the driver of the USB port itself shall be performed by not the installer of this operation gestalt but the installer of attachment in Windows like said operation gestalt. Moreover, if a USB driver is installed, it will create a new port.

[0057] The above actuation is the general specification of the USB driver in Windows. Once a USB driver is installed, the installer of this operation gestalt will recognize a new port, will obtain the connected printer name, and will choose the optimal printer driver. And during the copy of a file, the screen of [drawing 8](#) is displayed, file copy processing is performed, and a printer driver is registered. A user should make only starting of an installer, and connection of a printer.

[0058] The hardware configuration for realizing this invention is shown in [drawing 6](#) . Since the configuration of [drawing 6](#) is the same as that of [drawing 1](#) of said operation gestalt almost, the explanation which overlapped here is omitted.

[0059] In [drawing 6](#) , a sign H07 is a printer interface and H08 is a printer. USB shall be used as a printer interface H07.

[0060] Next, actuation of the installer of the printer device driver of this operation gestalt is explained using the flow chart of [drawing 14](#) and [drawing 15](#) . The name of the execution file of a printer installer is SETUP.EXE like the above-mentioned 1st operation gestalt, and is in the condition stored in the hard disk, or is supplied by media, such as CD-ROM and a floppy disk, like the above-mentioned operation gestalt.

[0061] If the binary (SETUP.EXE) of the installer on these media is clearly started by the user or it is started via a wizard, in step S101 of drawing 14, the file related to a USB driver will be copied to a system (refer to drawing 2). It becomes unnecessary for a user to specify a file by copying the file of a USB driver beforehand.

[0062] Step S102 is a step which enumerates all the port names that a system has and is memorized in memory as a port list A. The structure of a port list is shown in drawing 9. Drawing 9 is the example which memorized a port called LPT1, COM1, FILE, and USBPRN01.

[0063] In step S103, a user is urged to connect a USB printer in this screen at the step which displays the installer screen of drawing 7. If a user connects a printer with a USB port, the processing for which it waits until it becomes the interval time amount of a port check in step S104 will be put in. This interval time amount is made into 2 seconds with this operation gestalt. If it has been 2 seconds, it will progress to step S105.

[0064] At step S105, all the port names that a system has are enumerated and it memorizes in memory as a port list B.

[0065] At step S106, the port which exists in the port list A is deleted from the port list B. The newly added port remains in the port list B as a result of this deletion. If the port is not added, nothing remains.

[0066] At step S107, it judges whether the port remains in the port list B. If the port does not remain and return and a port remain in step S104, it will progress to step S108.

[0067] At step S108, the port name which remains in the port list B judges whether it is the port of a USB interface. With this operation gestalt, the port name of a USB interface serves as a "USBPRN figure." Then, the head of the port name which remains in the port list B judges whether it is "USBPRN", and if a head is "USBPRN", it is judged that it is the port of a USB interface. If it judges that it is the port of a USB interface, it will progress to step S109. Otherwise, it returns to step S104.

[0068] At step S109, the port of a USB interface is opened, in step S110, the GET\_DEWCE\_ID request of a USB protocol is required and the device ID of the connected USB device is acquired from the opened port. This device ID is a character string as shown in drawing 10. As for the device ID of drawing 10, the product name shows [ the manufacturer of a device ] "ABC-3000" by "Qanon".

[0069] At step S111 of drawing 15, the port of the USB interface opened at step S109 is closed.

[0070] Step S112 is a step which makes plug and play ID (omitting PnPID) from a device ID. This PnPID adds the name of the MFG tag of a device ID, and the name of a MDL tag. In the case of the device ID of drawing 10, PnPID is set to "QancmABC-3000."

[0071] At step S113, it judges whether PnPID created at step S112 in the correspondence printer list which an installer has is indicated. Here, the structure of the correspondence printer list of these operation gestalten is shown in drawing 11. A correspondence printer list is constituted from PnPID and a corresponding file name by drawing 11 so that it may be shown. In the example of drawing 11, it turns out that this installer deals with 3 models, "ABC-300" made from "Qanon", "ABC-80v", and "ABC-800alpha." If PnPID created at step S112 is indicated by the correspondence printer list, and it is judged as a correspondence printer, it progresses to step S114 and it is not indicated, it returns to step S104.

[0072] The install screen 1 displayed at step S103 is eliminated, and the install screen 2 of drawing 8 is expressed as step S114 in step S115 during the copy of a file. At step S116, the file name indicated by the correspondence printer list is acquired. At step S117, the file acquired at step S116 is copied.

[0073] At step S118, it judges whether all the files of a correspondence printer list were copied. If all copies are completed, it will progress to step S119. If the copy is not completed, it returns to step S117.

[0074] At step S119, it registers with a system by making into a printer driver the file copied at step S117. At step S120, the install screen 2 displayed at step S115 is eliminated.

[0075] Since the increment judging in a port which judges whether input/output port increased in number as mentioned above performs, and it is made performing install processing when the printer name which acquired the printer name and was acquired judges in the printer by which an installer corresponds through the port which increased and is judged to be a correspondence model, when a device like a printer connects also with the interface with which a port name like a USB interface had not been decided beforehand, the optimal driver can install.

[0076] A [operation gestalt of \*\* 3rd] book operation gestalt is an example which judges the addition of a port more efficiently than the previous operation gestalt 2. A hardware configuration is made into the same thing as the above-mentioned operation gestalt, and actuation of the installer in this operation gestalt is hereafter explained using the flow chart of drawing 16 and drawing 17.

[0077] The name of the execution file of a printer installer is SETUP.EXE like the above-mentioned 2nd operation gestalt, and is in the condition stored in the hard disk, or is supplied by media, such as CD-ROM and a floppy disk, like the above-mentioned operation gestalt.

[0078] If the binary (SETUP.EXE) of the installer on these media is clearly started by the user or it is started via a wizard, in step S201 of drawing 16, the file related to a USB driver will be copied to a system. It becomes unnecessary for a user to specify a file by copying the file of a USB driver beforehand.

[0079] The installer screen of drawing 7 is expressed as step S202. A user is urged to connect a USB printer on this screen. If a user connects a USB printer, in step S203, processing which waits for the message sent from a system will be performed. Here, a message is an instruction sent from a system to application. For example, a message is sent when the message and key which rewrite a screen are pressed.

[0080] At step S204, the message sent from the system judges whether it is a WM\_DEVICECHANGE message. This WM\_DEVICECHANGE message is sent when the condition of a device connected to the personal computer is changed. If a message is a WM\_DEVICECHANGE message here, it progresses to step S205, otherwise, it returns to step S203.

[0081] At step S205, the factor of a WM\_DEVICECHANGE message is investigated and it judges whether the port was added or not. If a port is added, it will progress to step S206. Otherwise, it returns to step S203.

[0082] At step S206, the head of the name of the added port judges whether it is "USBPRN". When step S206 is affirmed, it progresses to step S207, and when denied, it returns to step S203. Step S207 is a step which opens the port judged to be USB at step S206. Step S208 requires the GET\_DEVICE\_ID request of a USB protocol from the opened port, and acquires a device ID. This TEBAISU ID is the same as the previous operation gestalt 2. Step S209 closes the port of the USB interface opened at step S207.

[0083] Plug and play ID (omitting PnPID) is made from step S210 from a device ID. PnPID adds the name of the MFG tag of a device ID, and the name of a MDL tag as above-mentioned.

[0084] At step S211, it judges whether PnPID created at step S210 in the correspondence printer list which an installer has is indicated. This correspondence printer list is the same as the previous operation gestalt 2. If PnPID created at step S210 is indicated by the correspondence printer list, and it is judged as a correspondence printer, it progresses to step S212 and it is not indicated, it returns to step S203.

[0085] At step S212, the install screen 1 displayed at step S202 is eliminated, and the install screen 2 of drawing 8 is expressed as step S213. And at step S214, the file name indicated by the correspondence printer list is acquired. Furthermore, it is the step which copies



the file acquired at step S214 in step S215.

[0086] It judges whether step S216 copied all the files of a correspondence printer list, and if all are copied, it will progress to step S217. Step S217 is registered into a system by making into a printer driver the file copied at step S215. Furthermore, at step S218, the install screen 2 displayed at step S213 is eliminated.

[0087] The message sent from the system at the time of device connection as mentioned above The increment judging in a port which judges whether it is a WM\_DEVICECHANGE message and judges whether input/output port increased in number according to the contents further is performed. Since it is made to perform install processing when the printer name which acquired the printer name and was acquired judges in the printer by which an installer corresponds through the port which increased and is judged to be a correspondence model The optimal driver can be installed when a device like a printer is connected also with the interface with which a port name like a USB interface was not decided beforehand. Since the message sent from the system at the time of device connection with this operation gestalt is used, the addition of a port can be judged more efficiently than the previous operation gestalt 2.

[0088] The example which judges the optimal printer using further different information below in [the 4th operation gestalt] is shown. Since the information used with this operation gestalt is the information which the USB device surely has, it is also possible to apply to the installer of devices other than a printer. A hardware configuration is made into the same thing as the above-mentioned operation gestalten 2 and 3, and actuation of the installer in this operation gestalt is explained using the flow chart of drawing 18 and drawing 19.

[0089] The name of the execution file of a printer installer is SETUP.EXE like the above-mentioned 3rd operation gestalt, and is in the condition stored in the hard disk, or is supplied by media, such as CD-ROM and a floppy disk, like the above-mentioned operation gestalt.

[0090] If the binary (SETUP.EXE) of the installer on these media is clearly started by the user or it is started via a wizard, it will progress to step S301 of drawing 18.

[0091] At step S301, the file related to a USB driver is copied to a system. It becomes unnecessary for a user to specify a file by copying the file of a USB driver beforehand.

[0092] At step S302, all the port names that a system has are enumerated and it memorizes in memory as a port list A. The structure of a port list is as having been shown in drawing 9, and a port called LPT1, COM1, FILE, and USBPRN01 in drawing 9 is memorized here.

[0093] Step S303 is a step which displays the installer screen of drawing 7. A user is urged to connect a USB printer on this screen. And it is the processing for which it waits until it becomes the interval time amount of a port check at step S304. With this operation gestalt, interval time amount is made into 2 seconds like the 2nd operation gestalt, and if it has been 2 seconds, it will progress to step S305.

[0094] At step S305, all the port names that a system has are enumerated and it memorizes in memory as a port list B. It is the step which deletes the port which exists in the port list A from the port list B at step S306. As a result of deleting, the newly added port remains in the port list B. If not added, nothing remains.

[0095] Step S307 is a step which judges whether the port remains in the port list B. If the port does not remain, it returns to step S304. Moreover, if the port remains, it will progress to step S308.

[0096] Step S308 is a step which the port name which remains in the port list B judges in the port of a USB interface. Since the port name of a USB interface serves as a "USBPRN figure" with this operation gestalt, the head of the port name which remains in the port list B judges whether it is "USBPRN", and if a head is "USBPRN", it is judged that it is the port of a USB interface. In the case of the port of a USB interface, it progresses to step S309, otherwise, it returns to step S304.

[0097] Step S309 opens the port of a USB interface. At step S310, a GET\_DESCRIPTOR request is required using a USB protocol from the opened port, and a device descriptor is acquired. This device descriptor is the structure which stored the fundamental information on a device, and is data not only common to a USB device but all Windows device files. The structure and the example of the contents of a device descriptor are shown in drawing 12. In drawing 12, the vendor ID of a device descriptor is ID which means a manufacturer, and Product ID is ID which distinguishes a product. In the case of drawing 12, in Vendor ID, Product ID is 0000 by 04 A9.

[0098] At step S311, the port of the USB interface opened at step S309 is closed. And at step S312, Vendor ID and Product ID are picked out from the device descriptor acquired at step S310.

[0099] At step S313, it judges whether the vendor ID taken out at step S312 in the correspondence printer list which an installer has, and Product ID are indicated. The structure of the correspondence printer list which the installer of this operation gestalt has is shown in drawing 13. A correspondence printer list consists of a vendor ID, a product ID, and a corresponding file name. If Vendor ID and Product ID which were created at step S312 are indicated by the correspondence printer list, it will be judged as a correspondence printer and will progress to step S314, and if unstated, it will return to step S304.

[0100] Step S314 eliminates the install screen 1 displayed at step S303. And the install screen 2 of drawing 8 is displayed in step S315. The install screen 2 is displayed during the copy of a file. Furthermore, step S316 acquires the file name indicated by the correspondence printer list, and step S317 copies the file acquired at step S316.

[0101] It judges whether step S318 copied all the files of the device driver of the printer of a correspondence printer list. If all copies are completed, it will progress to step S319, otherwise, it returns to step S317. And at step S319, it registers with a system by making into a printer driver the file copied at step S317, and the install screen 2 displayed at step S315 is eliminated in step S320.

[0102] A device descriptor is acquired using a GET\_DESCRIPTOR protocol as mentioned above. Acquire a device name (printer name) and the increment judging in a port which judges whether input/output port increased in number is performed. Since it is made to perform install processing when the printer name which acquired the printer name and was acquired judges in the printer by which an installer corresponds through the port which increased and is judged to be a correspondence model The optimal driver can be installed when a device like a printer is connected also with the interface with which a port name like a USB interface was not decided beforehand.

[0103] Although this operation gestalt is limited to a USB interface, if it is the interface to which a port is added when a device driver is installed, there is an advantage which can be carried out no matter what interface it may use.

[0104] Although Windows of Microsoft was considered as an OS above, when the mechanism similar to specifying devices, such as a printer, in OS of arbitration is used, it cannot be overemphasized that the above-mentioned technique is applicable.

[0105] With a [operation gestalt of \*\* 5th] book operation gestalt, when connecting external instruments, such as two or more printers, through the interface of a USB port, Ethernet, etc., the configuration for checking easily the external instrument which he wants to use is illustrated.

[0106] A "check command" is transmitted to the printer chosen by the host, and it enables it to check easily the printer which he wants to use by making a printer answer it with this operation gestalt. It is the configuration that the printer which received the "check



command" generates audible sound, and a user more specifically judges by this from which printer the sound sounded. Thereby, matching with the printer currently displayed by the host and an actual printer can be checked.

[0107] The hardware configuration for realizing this operation gestalt first is explained using drawing 20. Drawing 20 is a drawing equivalent to drawing 1 or drawing 6, and it is, and it carries out and it has given the same sign to the same corresponding member. Printer interfaces H07 shall be two or more sets of printers H81 like a USB port, and the port which can connect H82 – concurrently also here. Since the other configuration is the same as drawing 1 or drawing 6, detailed explanation is omitted here.

[0108] In this operation gestalt, although a "check command" is transmitted to a printer H81 and H82 – from a computer (host), a control procedure like the flow chart of drawing 21 and a dialog like drawing 22 are used in that case.

[0109] The control procedure of drawing 21 is CPU. It is in the condition stored in the hard disk as a program which can perform H01, or is supplied by media, such as CD-ROM and a floppy disk, like the above-mentioned operation gestalt. Such a program shall be supplied as a part of printer driver program, and, more specifically, shall be started via the print dialog of a control panel, the menu for printer control, or an icon to application. The title of a menu or an icon can consider things, such as "a printer check" and "printer selection."

[0110] If the above menus and icons start, this program will display in a list the printer name connected with the host in step S01 of drawing 21. If there are many printers, this field will be controlled by the viewing area for the middle left-hand side of drawing 22 to list the printer name connected with the host possible [scrolling]. The product name character string acquired using the GET\_DESCRIPTOR protocol as shown in drawing 12 can be used for a printer name, and also when the same printer exists, the character string which gave and generated the number automatically is further used for it after this product name character string, or using the nickname which the user set as arbitration is also considered.

[0111] Mouse click waiting from a user is performed at step S02. Here, by clicking a desired printer name with a mouse, a user chooses and transmits a check command by carrying out the depression of the "check" carbon button. Moreover, the printer used for printing can be chosen by carrying out the depression of the "selection" carbon button.

[0112] When a click event occurs, in step S03, it judges whether the part of the printer name of the dialog of drawing 22 was clicked. A click of a printer name performs step S04 and step S05. That is, step S04 returns the printer name which was carrying out inverse video until now, and carries out the inverse video of the newly clicked printer name at step S05.

[0113] Steps S06, S08, and S10 detect the click of each carbon button of "a check" of the dialog of drawing 22, "cancellation", and "selection", respectively.

[0114] First, at step S06, if it judges whether the "check" carbon button of drawing 22 was pushed and a "check" carbon button is pushed, in step S07, a check command will be sent to the printer under present inverse video. A format of this check command may be arbitrary, and as long as it can generate audible sound using the loudspeaker of a printer etc. with this operation gestalt, it may be what kind of thing. For example, it can be used if there are commands (command for beep sound generating etc.) corresponding to PDL (Page Description Language) of the printer etc. A printer will sound a beep sound, if this command is received.

[0115] When "cancellation" carbon button of drawing 22 is pushed at step S08, all printer selections are made into an invalid in step S09, and processing of a dialog is ended. When "selection" carbon button is pushed at step S10, it considers as the printer which newly chose the printer which is carrying out current inverse video at step S11, and processing of a dialog is ended. If "selection" carbon button is not pushed at step S10, it returns to step S02 and waits for a new mouse click.

[0116] On the other hand, in a printer side, processing like the flow chart of drawing 23 is performed. That is, at step S01, it judges whether it is the check command with which the command received at step S02 when receiving waiting of a command was performed and a command was received was transmitted at step S07 of drawing 21, and when it is a check command, a beep sound is sounded with step S03.

[0117] Printing processing which corresponds if it is processing of an original printer after step S04 and a receiving command is a printing command is performed (step S05).

[0118] Thus, since a user can investigate the list of printers using a dialog (drawing 22), can transmit a check command to the target printer and can generate a beep sound, he can check easily the printer which he wants to use.

[0119] When connecting two or more external instruments (for example, you may be input devices, such as a scanner) through the interface of not only a printer but a USB port, Ethernet, etc., the above-mentioned technique can be used in order to enable it to check easily the external instrument which he wants to use.

[0120] In addition, although the command which sounds a sound with this operation gestalt as a check command was used, since the essence of this invention is reporting that a user can check the printer of a printing place by making a printer execute the command which carries out different actuation from printing, the actuation performed by the printer according to a check command may be the actuation which blinks not only generating of audible sound but LED.

[0121] Contrary to the 5th operation gestalt of the [operation gestalt of \*\* 6th] above, by operating it by the external instrument (printer) side, considering as "selection" condition is also considered in order to use the external instrument for an output (or input).

[0122] For example, when the selection carbon button B02 is formed in the printer (it is equivalent to H81 of drawing 20, or H82) as an external instrument and a user wants to use that printer for an output as shown in drawing 26, the depression of this selection carbon button B02 is carried out. By the predetermined protocol, the depression of this selection carbon button B02 is transmitted to a host side, and it is controlled by the host side who received this in the "selection" condition so that it may use that printer for an output. In addition, the carbon button B01 of drawing 26 is a power button.

[0123] The hardware configuration by the side of the host (computer) of this operation gestalt is the same as that of the 5th operation gestalt, and good. In this operation gestalt, a dialog like a control procedure like the flow chart of drawing 24 and drawing 25 is used. What is necessary is just to start the program of drawing 24 by the same opportunity as the 5th operation gestalt. Since the whole processing configuration of drawing 24 is the same as that of drawing 21 almost, below about the same details as processing of drawing 21, explanation is omitted.

[0124] The printer connected now is expressed in the dialog of drawing 25 as step S01 of drawing 24.

[0125] At step S02, the status of all the printers that displayed the mouse input at step S01 by step S03 first when there were waiting and a mouse input is obtained. The "selection status" which shows whether the selection carbon button B02 of drawing 26 is pushed shall be contained in this status. As mentioned above, the depression of this selection carbon button B02 is transmitted to a host side by the predetermined protocol. If this selection status is 1, it means that the selection carbon button B02 was pushed by the printer of this \*\*.

[0126] It judges whether there is any printer which returned the status of "selection status = 1" at step S04, if there is a printer of selection status = 1, it will progress to step S05, and the display of a printer name which was carrying out inverse video until now is returned, and the display of the printer name further set to selection status = 1 at step S06 is reversed.

[0127] If it judges whether the mouse click generated step S07 from the mouse input acquired at step S02, and a mouse click occurs, it

will progress to step S08 and the mouse click will not have occurred, it returns to step S02.

[0128] Step S08 judges, and if it is clicked whether the printer name of the dialog of drawing 25 was clicked, it progresses to step S09 and it is not clicked, it progresses to step S11.

[0129] It is the step which reverses the display of a printer name by which returned the display of a printer name which was carrying out inverse video until now at step S09, and the mouse click was carried out at step S10. That is, it enables it to choose a printer by manual operation like drawing 21 here.

[0130] At step S11, it judges whether "cancellation" carbon button of the dialog of drawing 25 was pushed. If this carbon button is pushed, it will progress to step S12, and all selections of a printer are made into an invalid, and processing of a dialog is ended.

[0131] At step S13, it judges whether "selection" carbon button of drawing 25 was pushed, and if not pushed, it returns to step S02 and a new mouse input is received. On the other hand, when a confirmation button is pushed, it considers as the printer which newly chose the printer which is carrying out current inverse video in step S14, and processing of a dialog is ended.

[0132] In a printer side, the selection status can be determined by performing processing like the flow chart of drawing 27. Since what is necessary is for drawing 27 to explain the task which performs processing about the selection status, and just to operate processing of printer original as another task, explanation is omitted about processing of printer original here.

[0133] Step S01 of drawing 27 receives the depression of the manual operation button of a printer, and it judges whether the selection carbon button (B02 of drawing 26) was pushed at step S02. When a selection carbon button is pushed, it progresses to step S03, the selection status is set as 1, and when the selection carbon button is not pushed, in step S04, the selection status is set as 0. A printer holds this status, and when there is a status demand, it should just carry out HOSUTOHE transmission using a predetermined protocol.

[0134] Thus, since the printer is already in the inverse video condition if the selection carbon button is pushed by the printer side when the list of printers is investigated using a dialog (drawing 25), only by confirming the selection condition with "selection" carbon button, a user can check easily the printer which he wants to use, and can choose it.

[0135] When connecting two or more external instruments (for example, you may be input devices, such as a scanner) through the interface of not only a printer but a USB port, Ethernet, etc., the technique of this operation gestalt can also be used in order to enable it to check easily the external instrument which he wants to use.

[0136] Although the configuration which prepares "selection" carbon button of dedication in a printer was illustrated with this operation gestalt, you may make it use the combination of the depression of an online carbon button, other existing manual operation buttons, or two or more existing manual operation buttons for "selection" actuation, for example.

[0137] If [the 7th operation gestalt] or this operation gestalt turns on a printer, also "also making choose automatically by a host's printer driver" it, using the printer as a printing place will be considered. The hardware configuration of this operation gestalt is the same as the operation gestalt 1, and good.

[0138] In this operation gestalt, the dialog (it is the same as the 6th operation gestalt) of a control procedure like the flow chart of drawing 28 and drawing 25 is used. What is necessary is just to start the program of drawing 28 by the same opportunity as a 5th and 6 operation gestalt. Since the whole processing configuration of drawing 28 is the same as that of drawing 21 and 24 almost, below about drawing 21 and the same details as processing of 24, explanation is omitted.

[0139] In step S01 of drawing 28, the printer connected now is displayed in the dialog of drawing 25, and it waits for a mouse input at step S02.

[0140] At step S03, it judges whether there is any newly connected printer. Here, a printer should just come to appear on a USB bus and networks, such as Ethernet, with "connection." Therefore, also when a USB bus and the printer connected through Ethernet etc. change to the condition of power-source ON from the condition of power-source OFF, it can detect at this step. When the address of the printer on a bus or a network is supervised or a power source is specifically switched on by the power button B01 of drawing 26 etc., even if it transmits an explicit signal from a printer side, it can judge whether there is any printer "was connected" newly.

[0141] When there is a newly connected printer, in step S04, the display of a printer name which was carrying out inverse video on the dialog until now is returned, and the display of the printer name of the printer "was connected" newly at step S05 is reversed.

[0142] It judges whether the mouse click occurred from the mouse input acquired at step S02, and at step S06, if a mouse click occurs, it progresses to step S07, and if it has not generated, it will return to step S02.

[0143] At step S07, it judges, and if it is clicked whether the printer name of the dialog of drawing 25 was clicked, it progresses to step S08 and it is not clicked, it progresses to step S10.

[0144] At step S08, the display of a printer name which was carrying out inverse video until now is returned, and the display of a printer name by which the mouse click was carried out is reversed in step S09.

[0145] Whether "cancellation" carbon button of the dialog of drawing 25 was pushed judges step S10, and if pushed, it will progress to step S11. At step S11, all selections of a printer are made into an invalid and processing of a dialog is ended.

[0146] At step S12, it judges whether "selection" carbon button of the dialog of drawing 25 was pushed, and if not pushed, it returns to step S02 and waits for a new mouse input. When a confirmation button is pushed, it considers as the printer which newly chose the printer which is carrying out current inverse video in step S13, and processing of a dialog is ended.

[0147] Thus, since the printer which acted as powering on is in the inverse video condition when the list of printers is investigated using a dialog (drawing 25), only by confirming the selection condition with "selection" carbon button, a user can check easily the printer which he wants to use, and can choose it.

[0148] When connecting two or more external instruments (for example, you may be input devices, such as a scanner) through the interface of not only a printer but a USB port, Ethernet, etc., the technique of this operation gestalt can also be used in order to enable it to check easily the external instrument which he wants to use.

[0149]

[Effect of the Invention] While performing the I/O to the device concerned by controlling a peripheral-device device via the device driver of the device concerned according to this invention so that clearly from the above explanation In the storage which stored the electronic equipment by which the registry information about registration of a device driver is registered into a registry information storage means, its control approach, and its control program and in which computer reading is possible When the device driver of a predetermined peripheral device is installed in a system, Since the device ID information corresponding to the device driver which should be registered was retrieved with said registry information storage means, and the configuration which deletes the registry information concerned is adopted when a part of retrieved registry information is unknown There is outstanding effectiveness that a USB device can be installed by itself easily, without not needing actuation in which an explicit driver is deleted and needing large sum user support.

[0150] Or a peripheral-device device is connected through the input/output port which can connect two or more peripheral devices concurrently. In the storage which stored the electronic equipment which performs the I/O to the device concerned by controlling a

peripheral-device device via the device driver of the device concerned, its control approach, and its control program and in which computer reading is possible The device name of the peripheral device corresponding to the port is acquired through the port which increased when it judged whether said input/output port increased and said input/output port increased. By adopting the configuration which performs install processing of the device driver of the peripheral device concerned when the correspondence model of specification [ the acquired device name ] is shown Like a USB interface, a port name is not decided beforehand but there is outstanding effectiveness that the optimal driver can be installed when a device like a printer is connected, also with the input/output port which can connect two or more peripheral devices concurrently.

[0151] Or the electronic equipment which connects a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and performs the I/O to the device concerned, In the storage which stored the control approach and its control program and in which computer reading is possible A check command for a user to check the peripheral device which should be chosen as an I/O place is transmitted via said input/output port. By adopting the configuration which reports reception of this check command to a user at the peripheral-device side which received this check command Like a USB interface, a port name is not decided beforehand but there is outstanding effectiveness that a user can specify easily peripheral devices, such as a printer which he should use, also with the input/output port which can connect two or more peripheral devices concurrently.

[0152] Or the electronic equipment which connects a peripheral-device device through the input/output port which can connect two or more peripheral devices concurrently, and performs the I/O to the device concerned, In the storage which stored the control approach and its control program and in which computer reading is possible The status information which becomes settled according to actuation of said peripheral device or operating state is received from said peripheral-device device through said input/output port. By adopting the configuration which chooses a peripheral device as an I/O place according to the received status information The outstanding effectiveness that peripheral devices, such as a printer which he wants to use using the status which becomes settled according to actuation of a peripheral device or operating state, for example, the actuation condition of the actuation means of a peripheral device, the injection condition of a main power supply, etc., can be checked easily, and can be chosen is acquired.

---

[Translation done.]

## \* NOTICES \*

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

## DRAWINGS

[Drawing 2]

(図 2)

ファイル名	説明
USBDEV. SYS	デバイスドライバ本体
USBDEV. INF	設定ファイル
SETUP. EXE	インストーラプログラム

ファイル構成図

[Drawing 9]

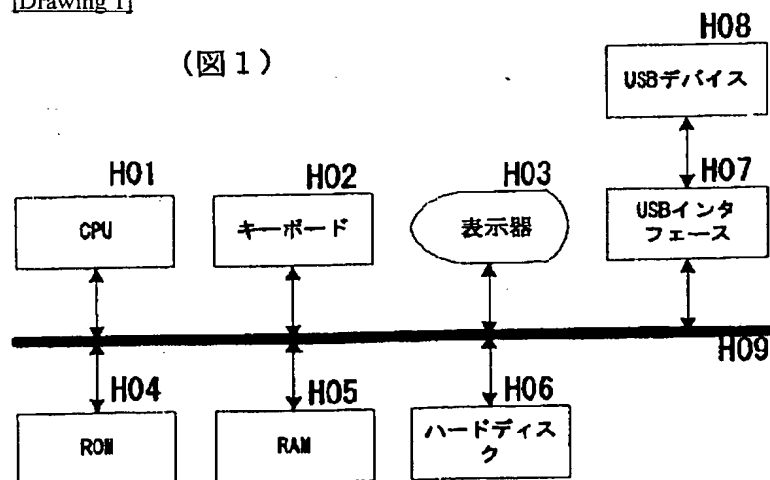
(図 9)

番号	ポート名
1	LPT1
2	COM1
3	FILE
4	USBPRN01

ポートリスト

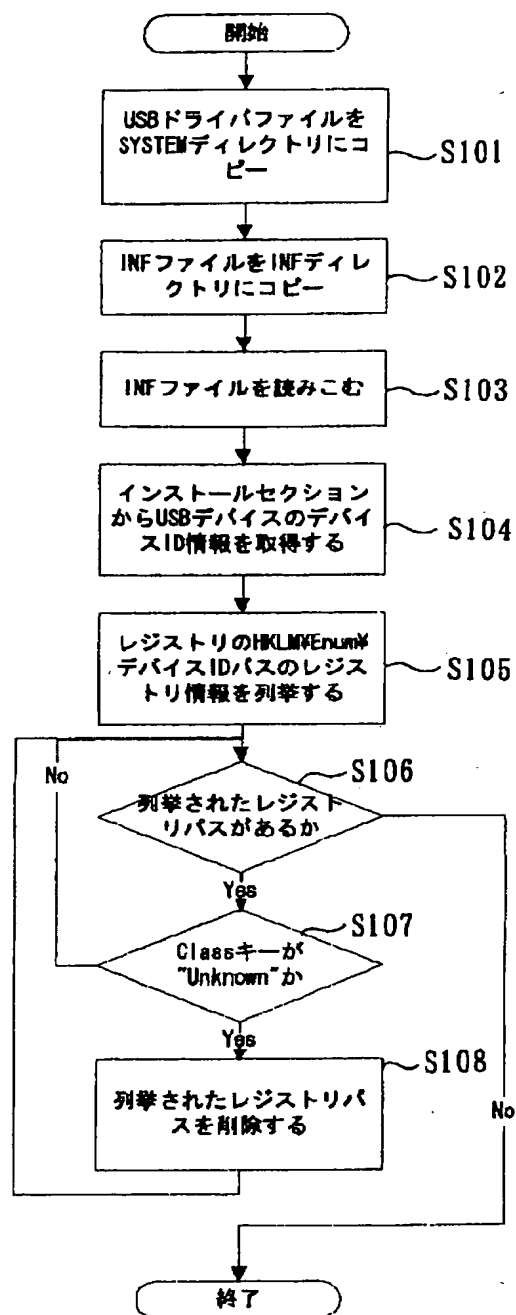
[Drawing 1]

(図 1)



ハードウェア構成図

[Drawing 5]

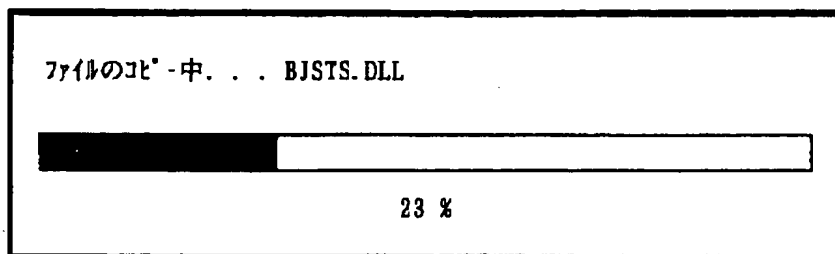


インストーラのフローチャート

(図 5)

[Drawing 8]

(図 8)



インストーラ画面 2

[Drawing 10]

(図 10)

MFG:Qanon;CMD:TEXT;MDL:ABC-3000;CLS:PRINTER;

デバイスID

[Drawing 3]

(図 3)

```
1. [Version]
2. Signature="$Windows 95$"
3. Class=USB
4. Provider=Qanon

5. [DestinationDirs]
6. DefaultDestDir=11

7. [Manufacturer]
8. Qanon=Qanon.Section

9. [Qanon.Section]
10. "Qanon USB Device"=Qanon.USB,USB\VID_04A9&PID_1052

11. [Qanon.USB]
12. CopyFiles=Qanon.CopyList
13. AddReg=Qanon.AddReg

14. [Qanon.CopyList]
15. usbdev.sys

16. [Qanon.AddReg]
17. HKR,,DevLoader,,*ntkern
18. HKR,,NTMPDriver,,usbdev.sys
19. HKR,,QanonSoft,,Printer
20. HKR,,Device,,USBPRN
21. HKR,,ID,,VID_04A9&PID_1052
22. HKR,,Instance,,Multiple
23. HKR,,DeviceDesc,, "Qanon USB Device"

24. [ControlFlags]
25. ExcludeFromSelect=USB\VID_04A9&PID_1052

26. [PreCopySection]
27. HKR,,NoSetupUI,,1

28. [SourceDisksNames]
29. 1 = "Qanon USB Device Disk",,0001

30. [SourceDisksFiles]
31. usbdev.sys = 1
```

## USBDEV. INFの内容

[Drawing 4]



レジストリ エディタ

レジストリ 編集(E) 表示(V) ヘルプ(H)

Enum

BIOS

ESDI

FLOP

HTRREE

INFRARED

ISAPNP

LPTENUM

MF

MONITOR

Network

PCI

Root

SCSI

USB

ROOT\_HUB

VID\_047E&PID\_1001

VID\_04A9&PID\_1052

000000

名前

データ

ad (標準)

14 00 00 00

Unknown

USB&CLASS\_07&SUBCLASS\_01&PROT\_02.USB&CLA

40 00 00 00

"Qanon USB Device"

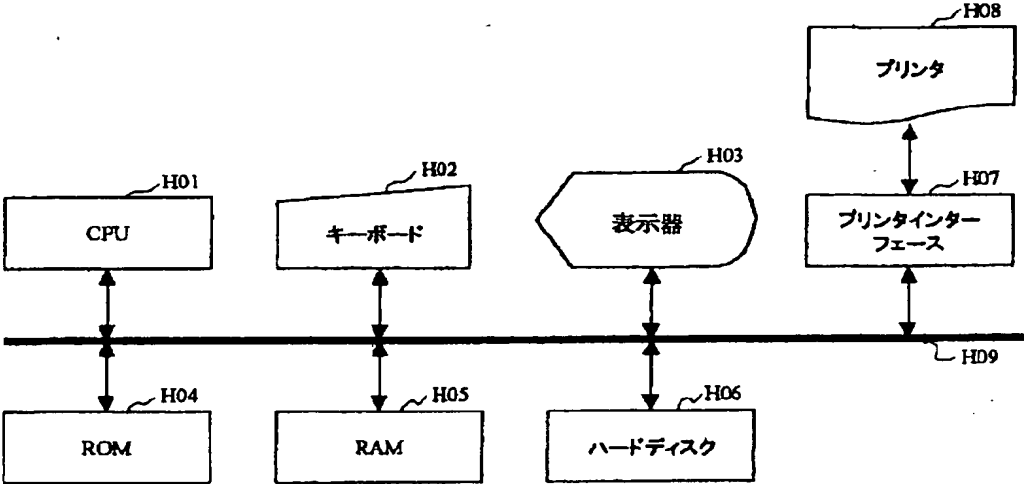
USB&VID\_04A9&PID\_1052&REV\_0103.USB&VID\_0

マイコンビュータKEY LOCAL MACHINE\Enum\USB&VID\_04A9&PID\_1052\000000

[Drawing 6]

レジストリ  
(図 4)

(図 6)



ハードウェアブロック図

[Drawing 11]

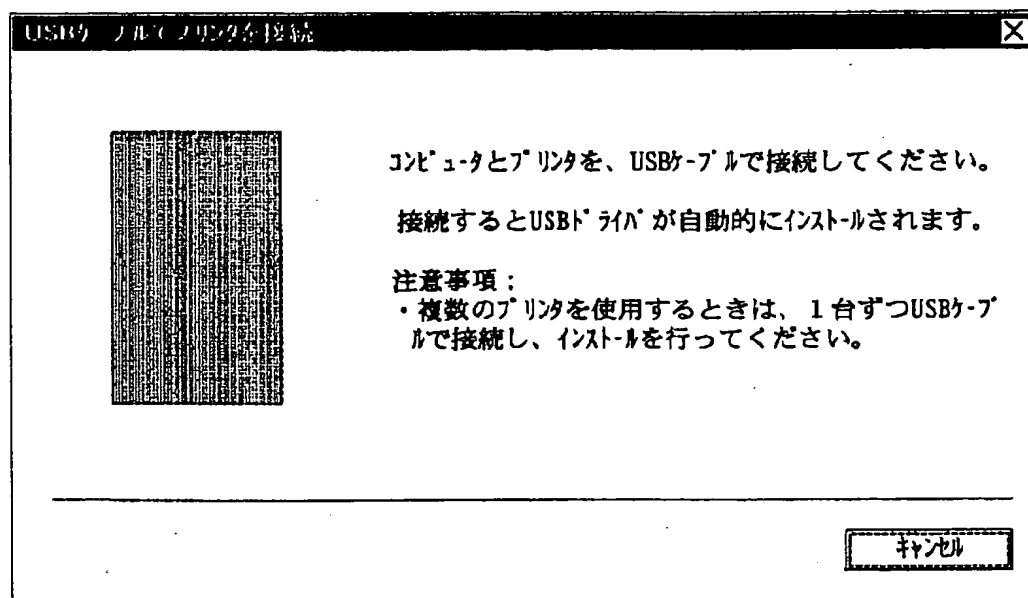
(図 1 1)

番号	PnP ID	必要ファイル
1	QanonABC-3000	ABC3000.DRV ABC3000.DLL QANON_UI.DLL QANON_UN.EXE
2	QanonABC-80v	ABC80V.DRV ABC80V.DLL QANON_UI.DLL QANON_UN.EXE
3	QanonABC-8000	ABC8000.DRV ABC8000.DLL QANON_UI.DLL QANON_UN.EXE

対応プリンタリスト

[Drawing 7]

(図 7)



インストーラ画面 1

[Drawing 12]

(図 1 2)

フィールド名	値 (16 進数)	意味
bLength	18	デバイスデスク립タのバイトサイズ
bDescriptorType	01	デバイスデスク립タのタイプ番号
bcdUSB	0100	USB バージョン番号
bDeviceClass	00	デバイスクラスコード
bDeviceSubClass	00	デバイスサブクラスコード
bDeviceProtocol	00	デバイスプロトコルコード
bMaxPacketSize	08	最大パケットサイズ
idVender	04A9	ベンダーID
idProduct	0000	プロダクト ID
bcdDevice	0100	デバイスバージョン番号
iManufacturer	01	ベンダー名文字列番号
iProduct	02	プロダクト名文字列番号
iSerialNumber	03	シリアル番号文字列番号
bNumConfigurations	01	コンフィグレーション数

デバイスデスク립タの構造

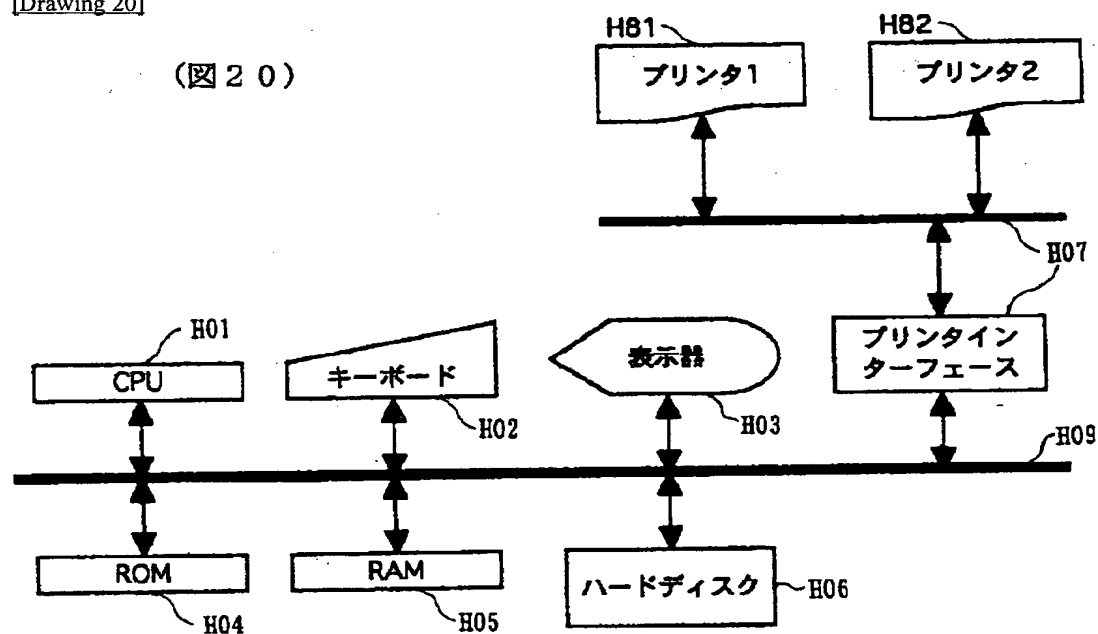
[Drawing 13]

(図 13)

番号	ベンダーID	プロダクトID	必要ファイル
1	04A9	0001	ABC3000.DRV ABC3000.DLL QANON_UI.DLL QANON_UN.EXE
2	04A9	0002	ABC80V.DRV ABC80V.DLL QANON_UI.DLL QANON_UN.EXE
3	04A9	0003	ABC8000.DRV ABC8000.DLL QANON_UI.DLL QANON_UN.EXE

実施形態3の対応プリンタリスト

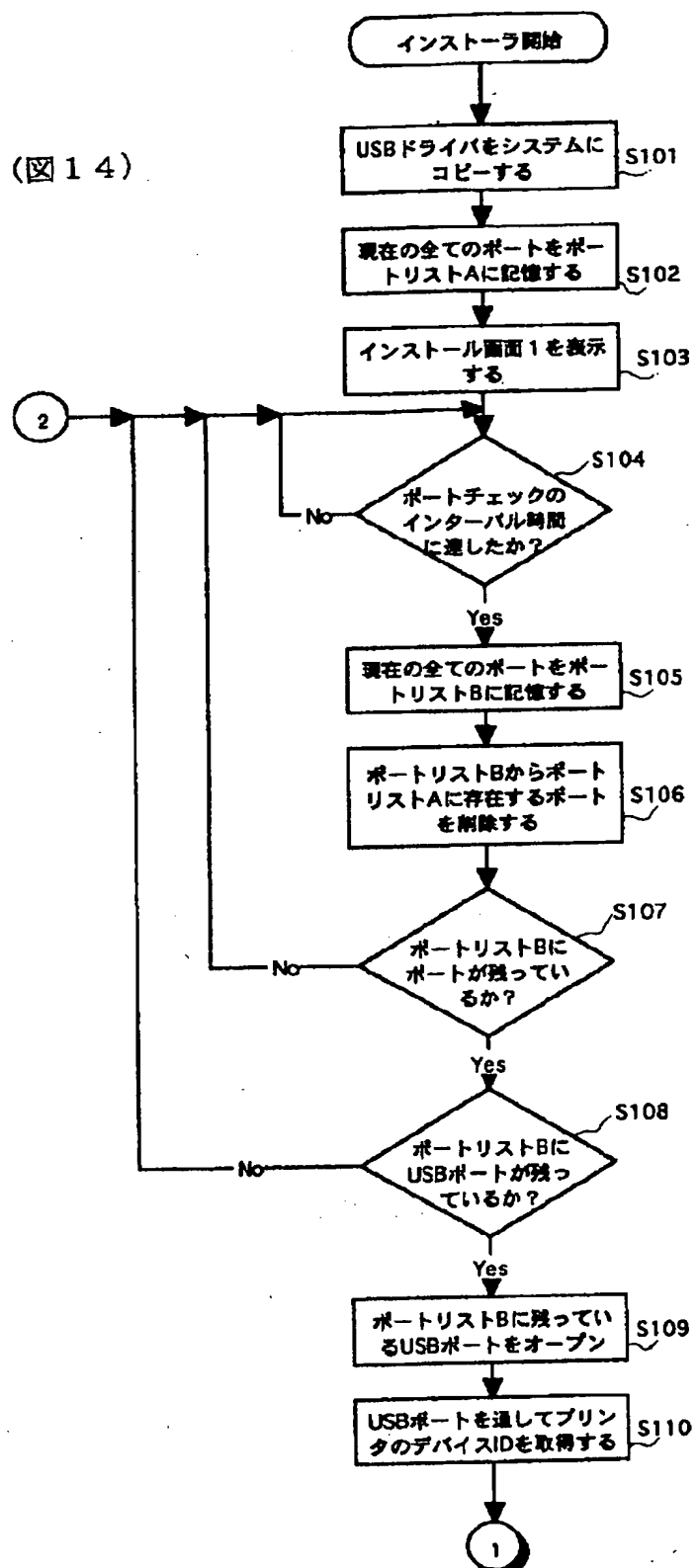
[Drawing 20]



ハードウェアブロック図

[Drawing 14]

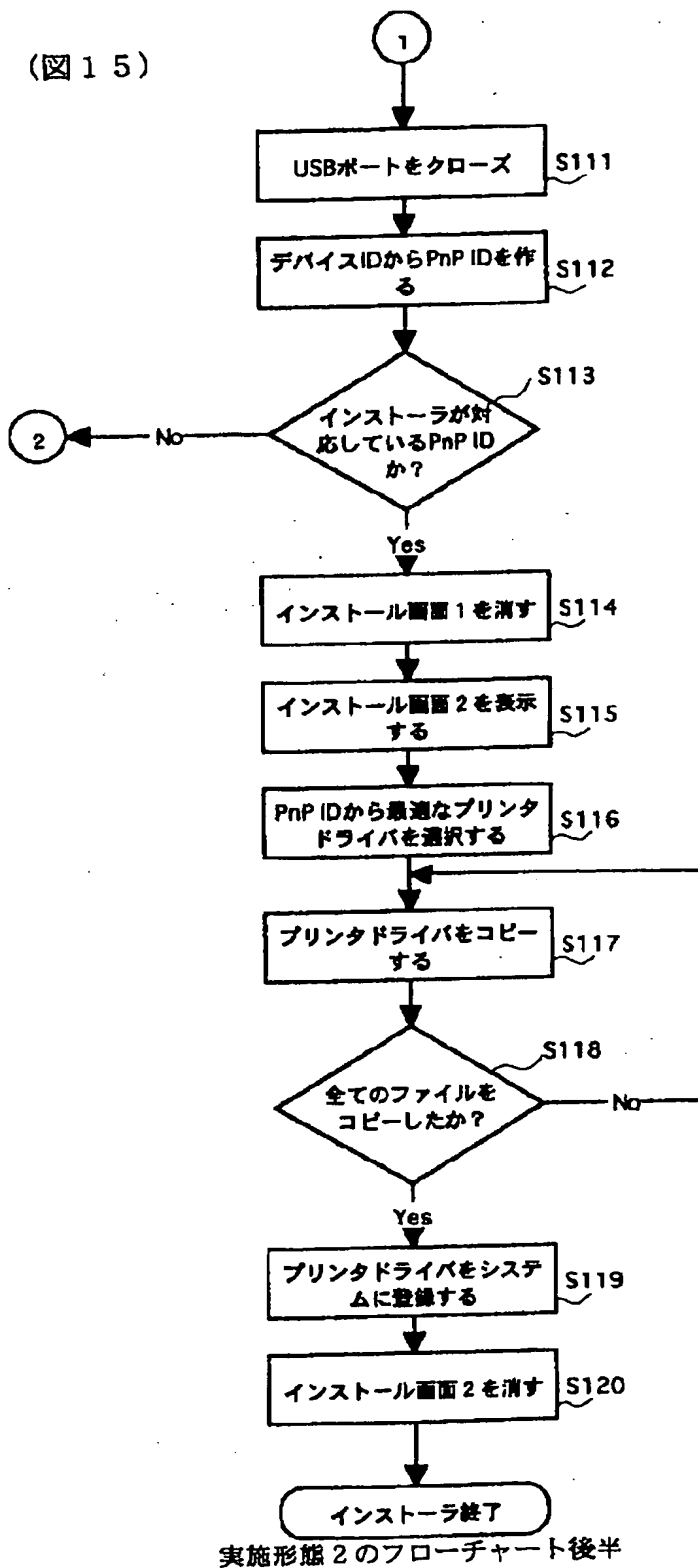
(図 14)



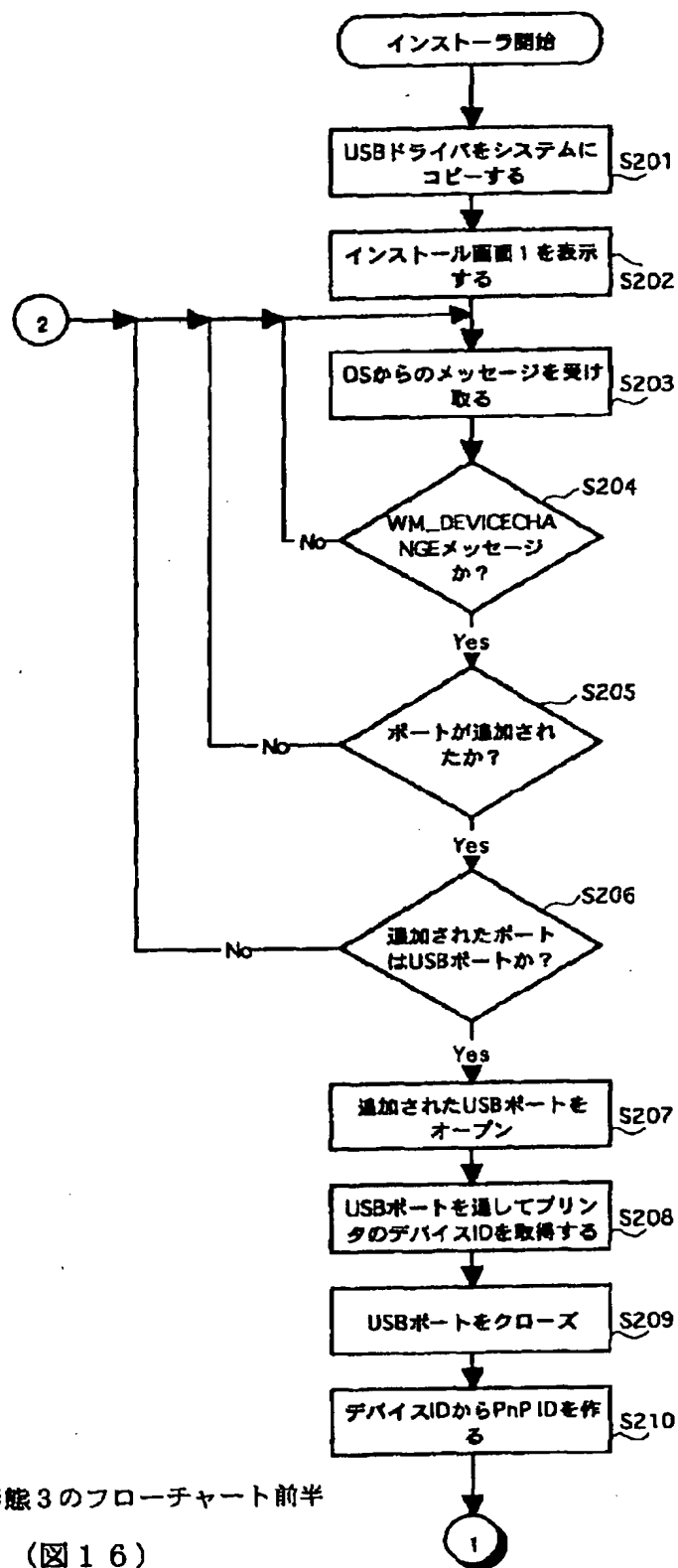
実施形態2のフローチャート前半

[Drawing 15]

(図 15)



[Drawing 16]



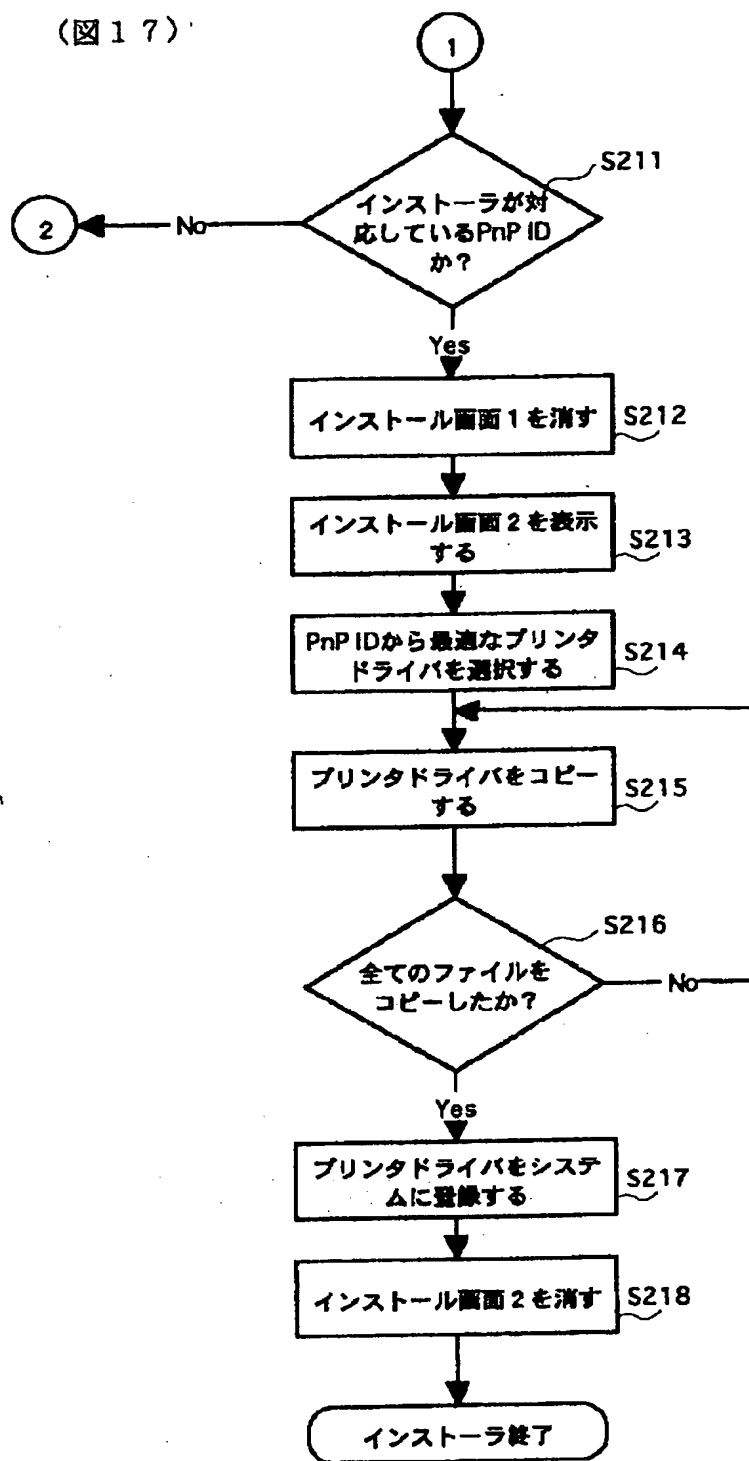
実施形態3のフローチャート前半

(図16)

[Drawing 17]

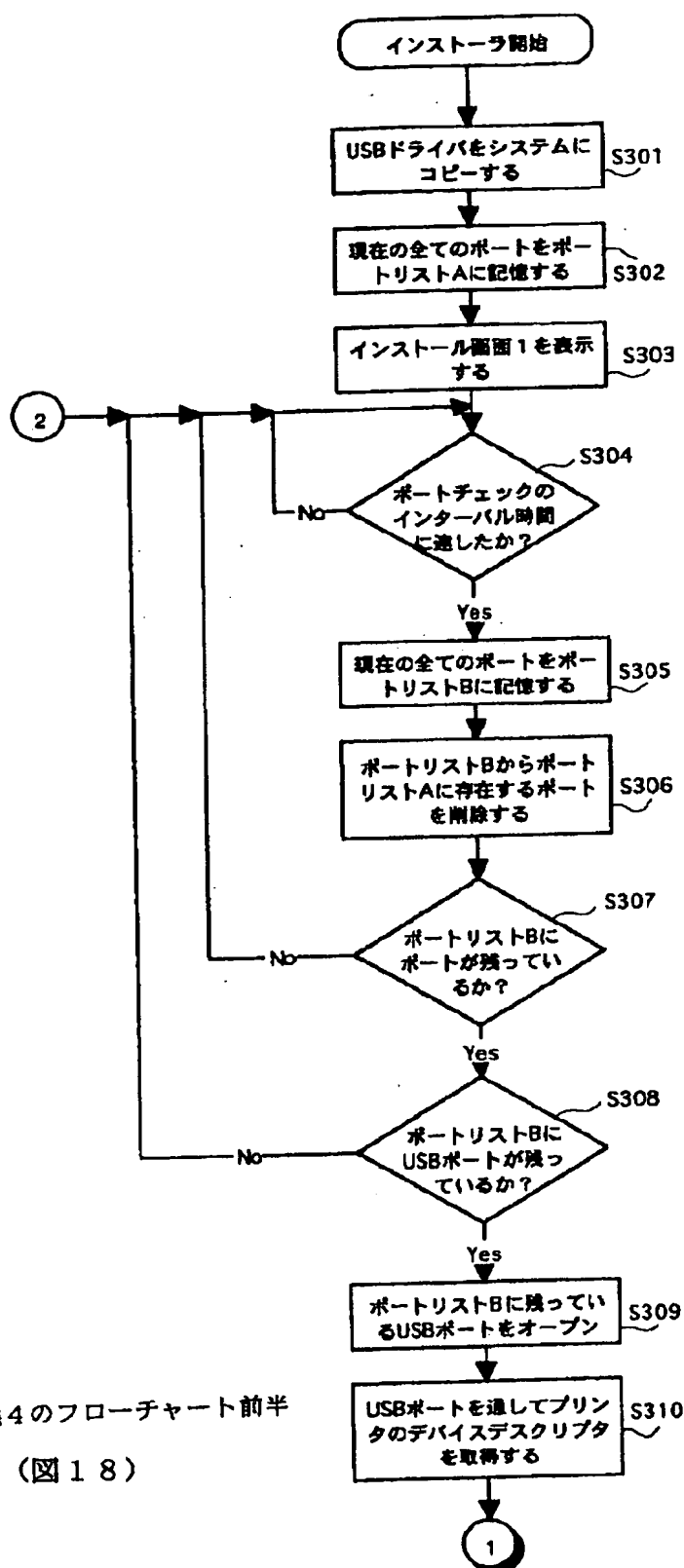


(図 17)



実施形態3のフローチャート後半

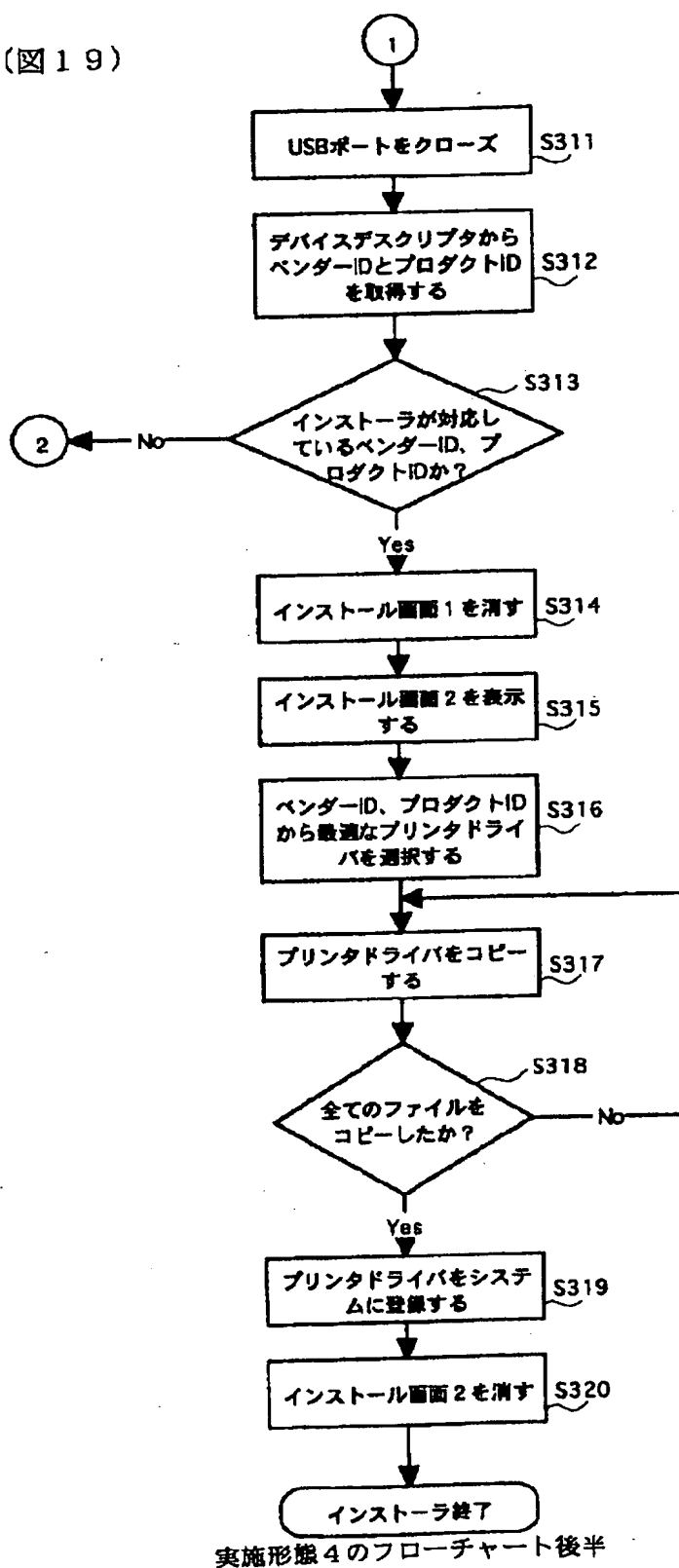
[Drawing 18]



実施形態4のフローチャート前半  
(図18)

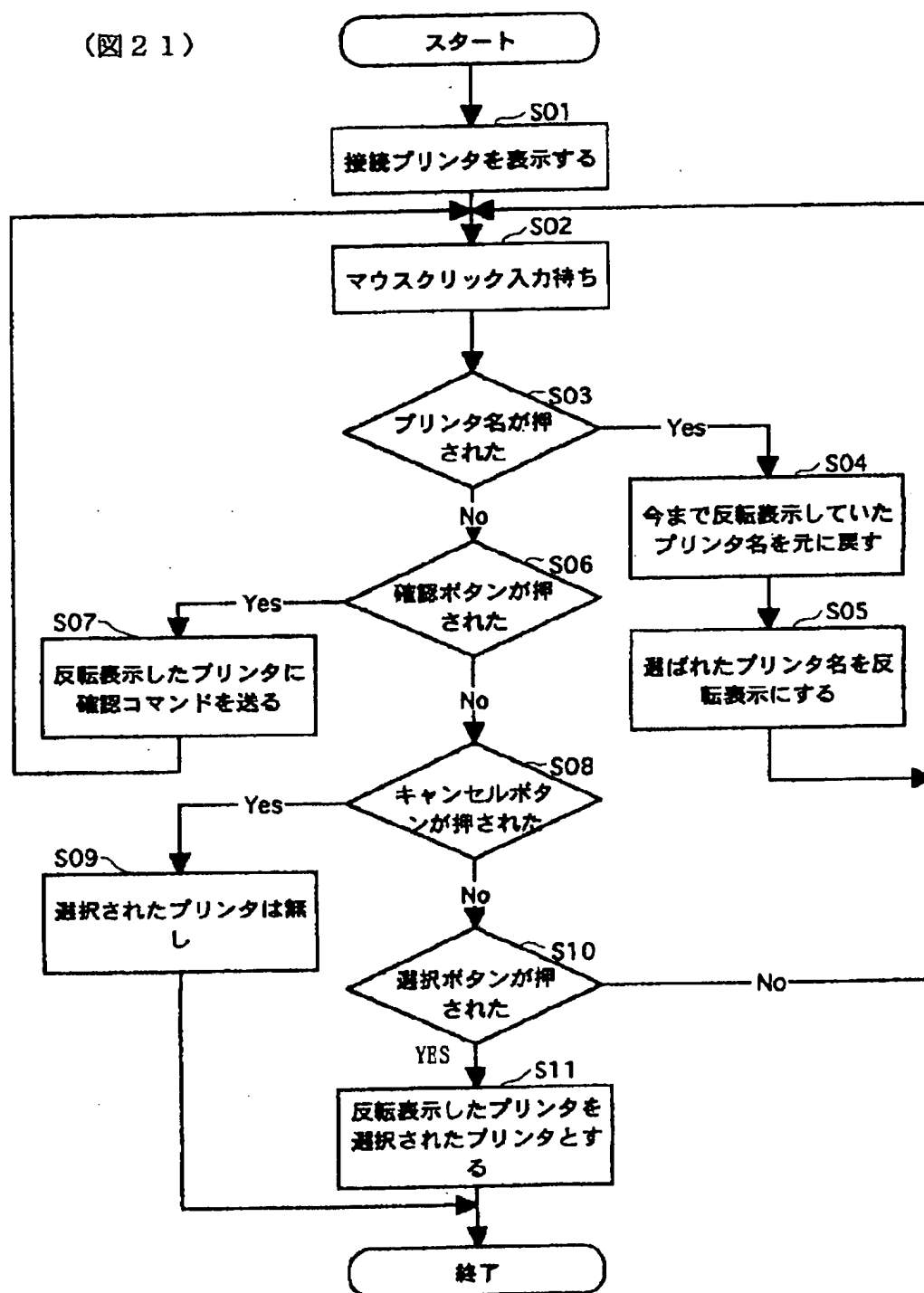
[Drawing 19]

(図 19)



[Drawing 21]

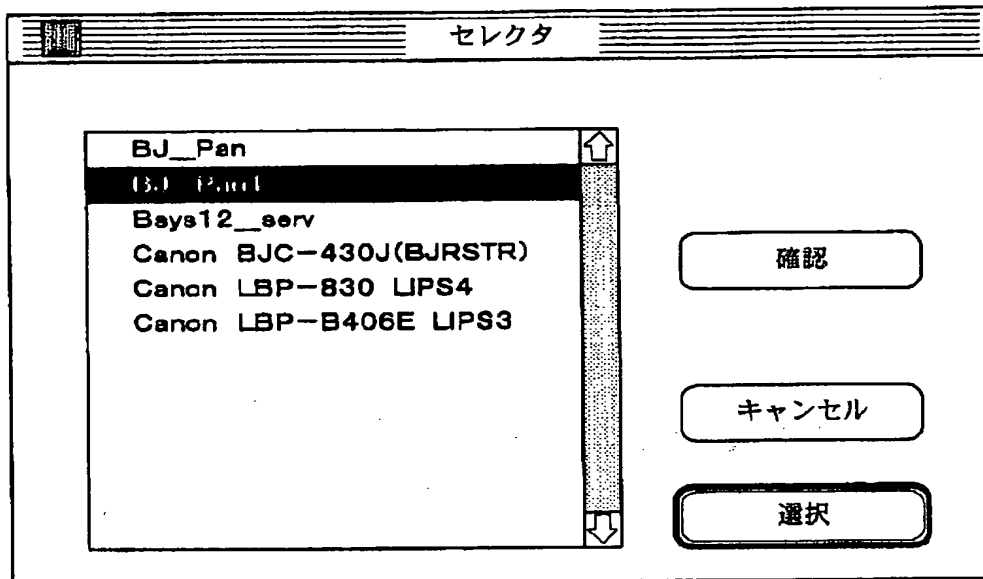
(図 21)



実施形態 5 のフローチャート

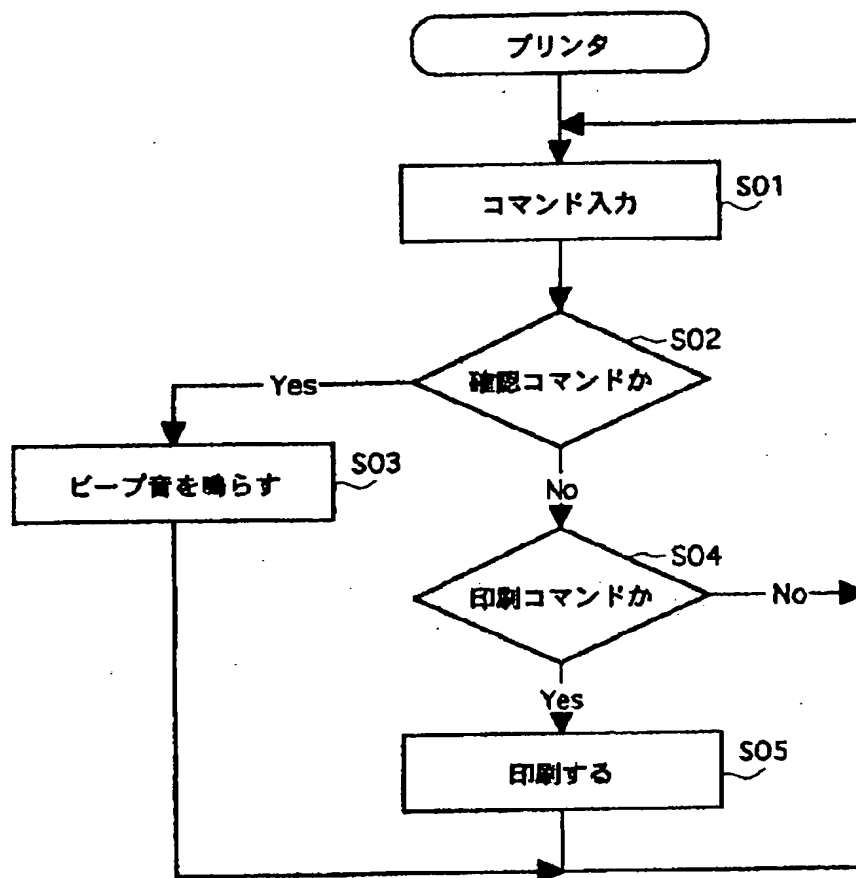
[Drawing 22]

(図 22)



実施形態5のプリンタ選択ダイアログ

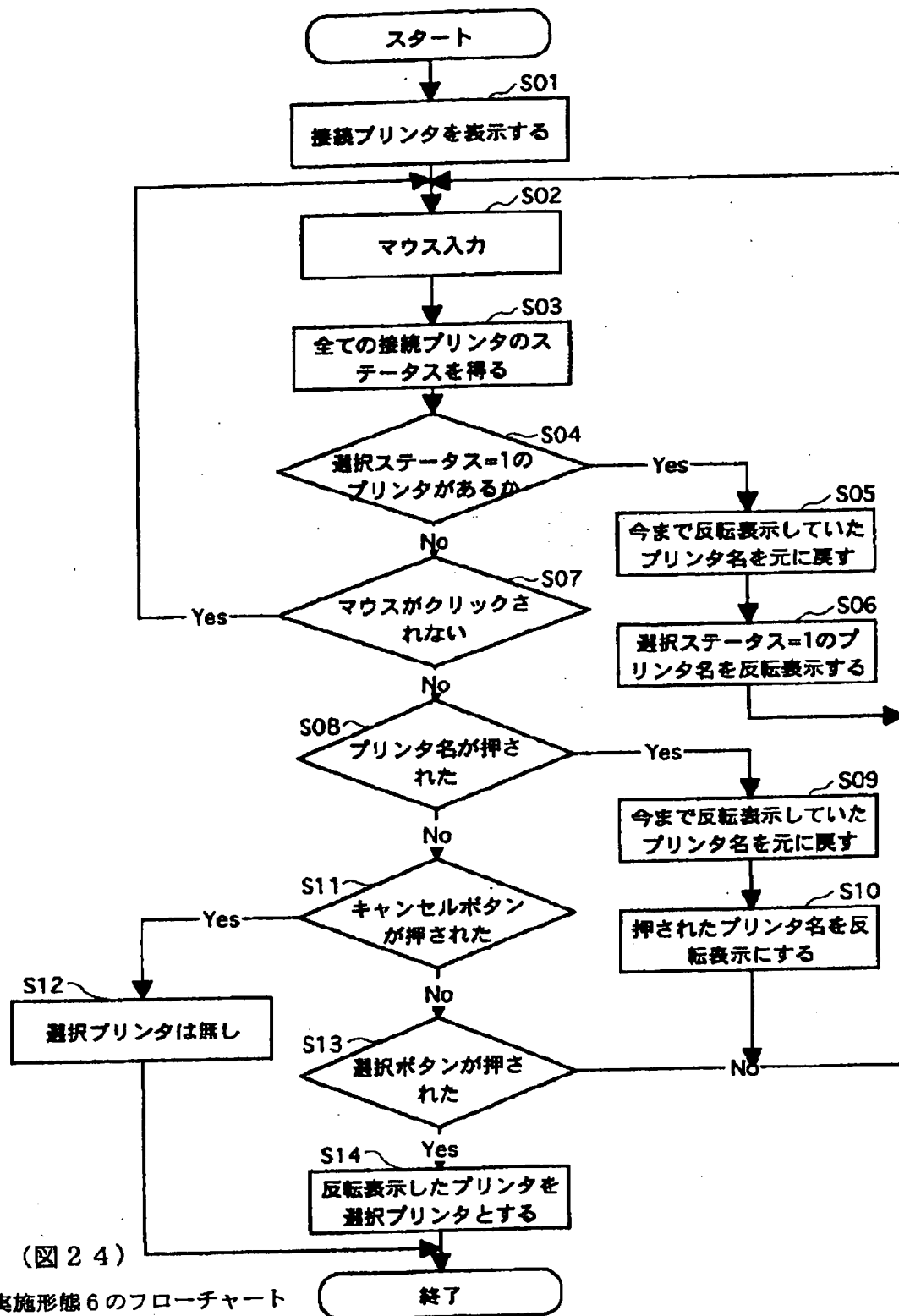
[Drawing 23]



プリンタのフローチャート

(図 23)

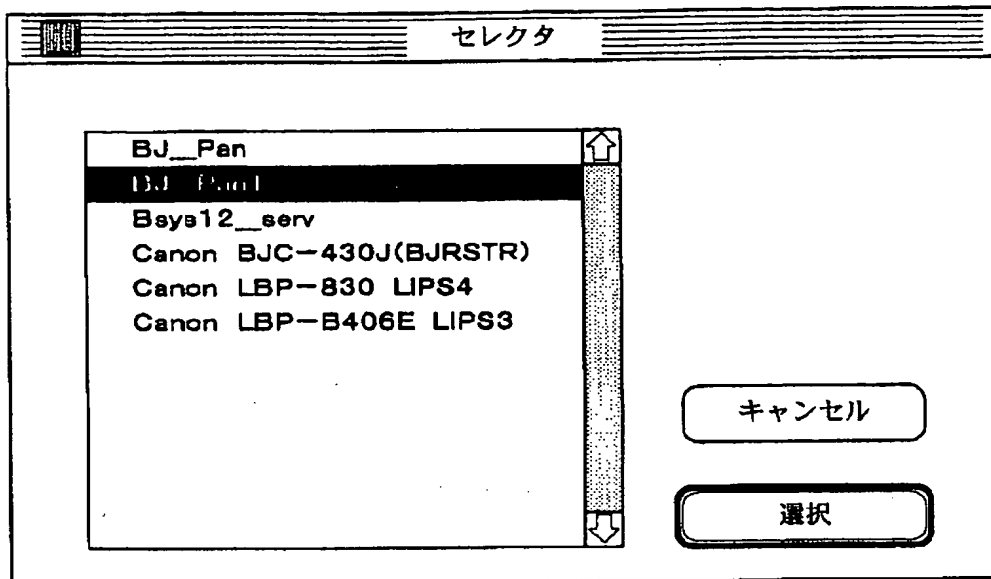
[Drawing 24]



[Drawing 25]



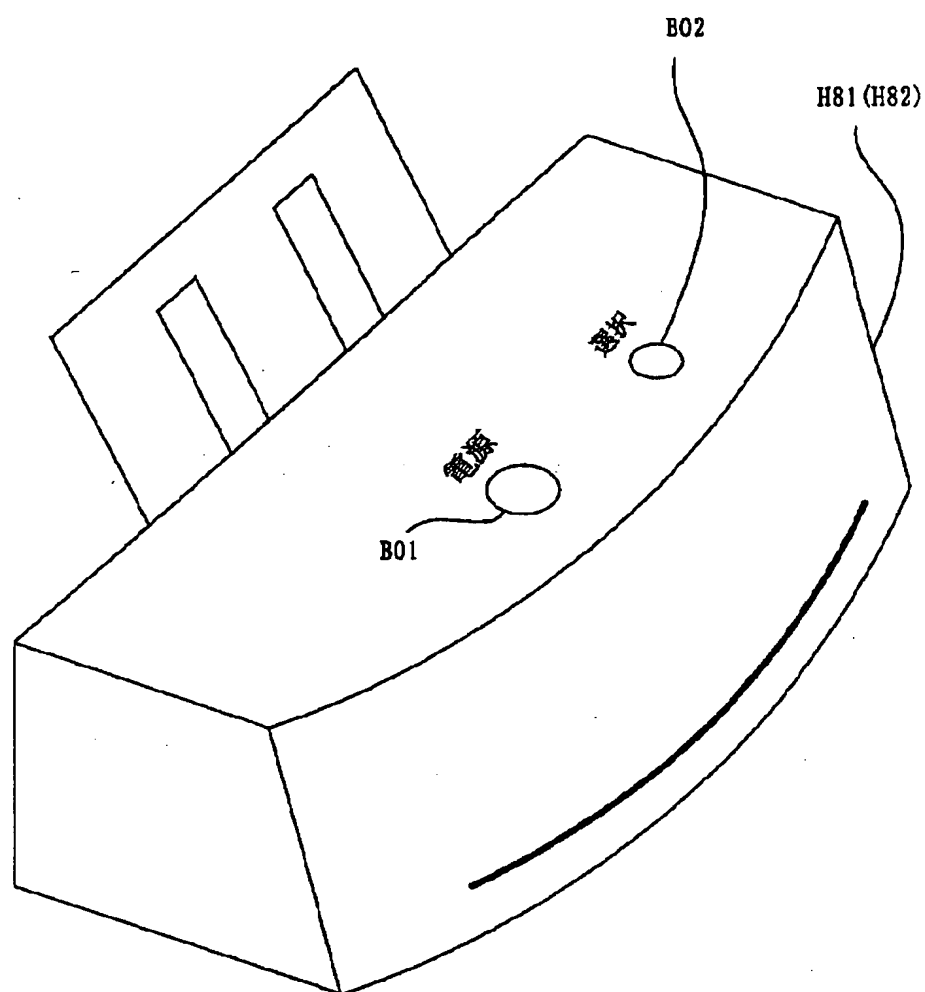
(図 25)



実施形態 6 のプリンタ選択ダイアログ

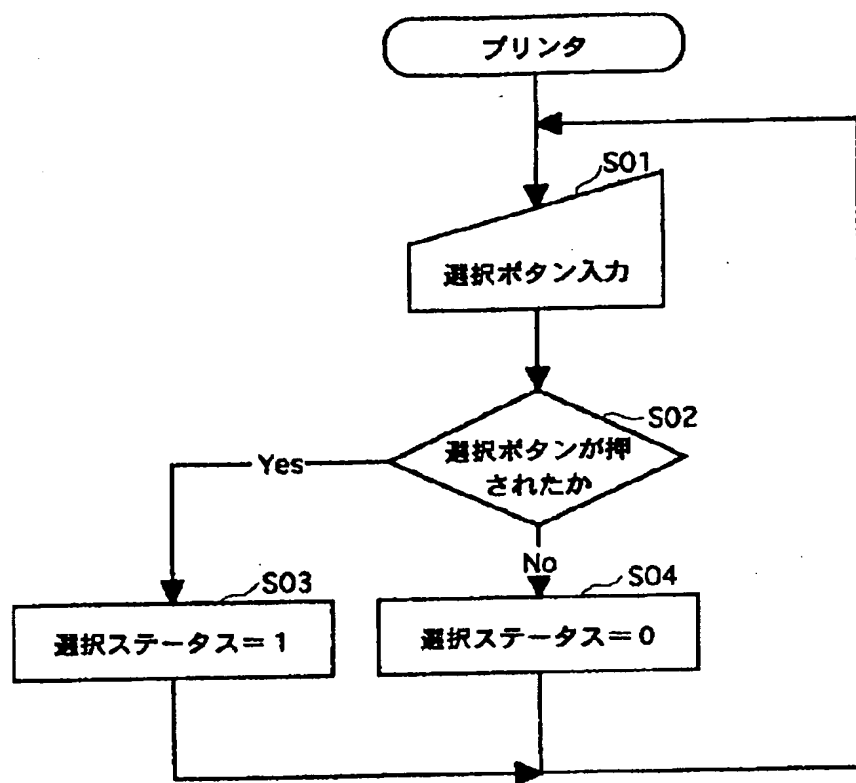
[Drawing 26]

(図 26)



実施形態 6 の選択ボタン付きプリンタ

[Drawing 27]

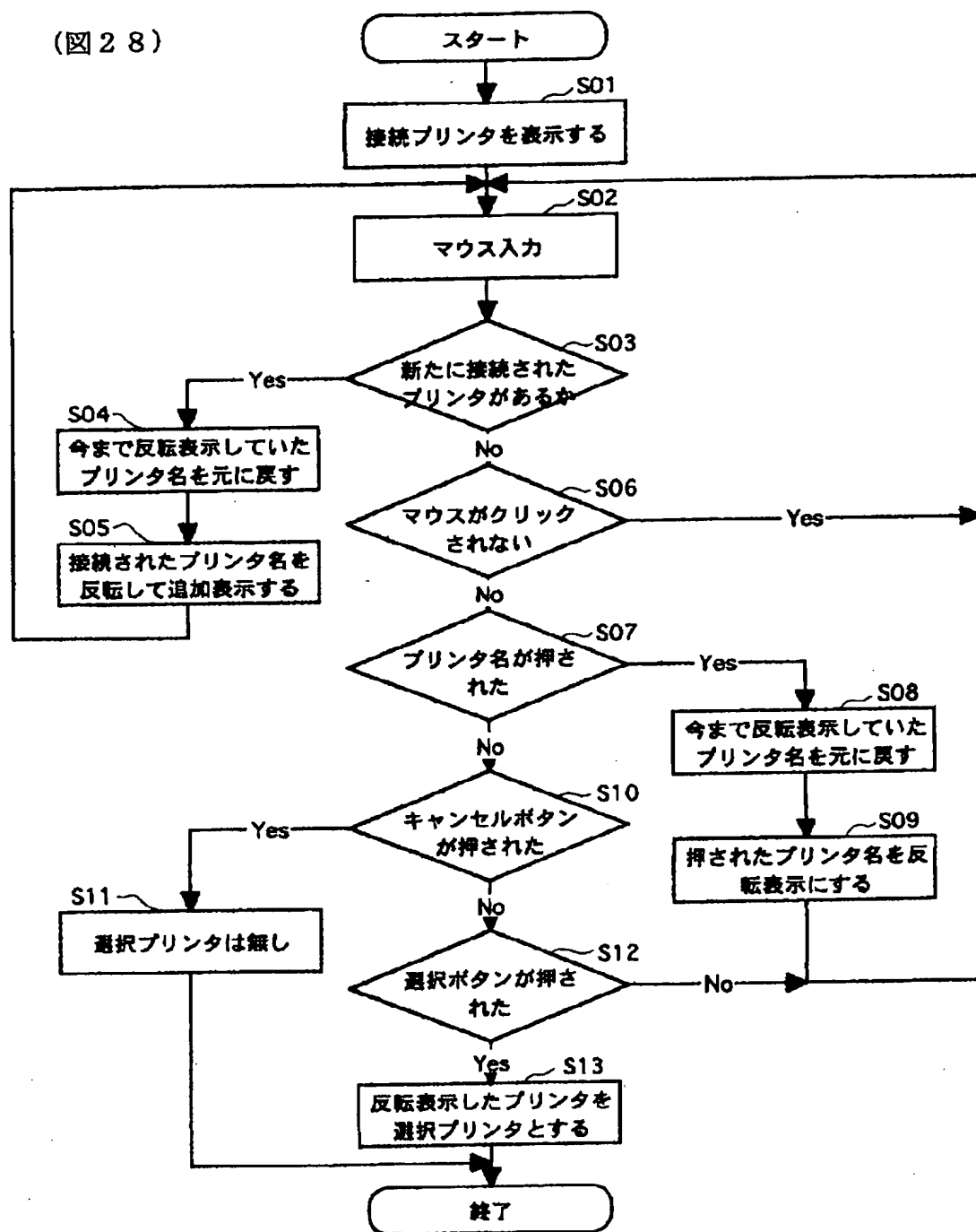


選択ステータスのフローチャート

(図 27)

[Drawing 28]

(図 28)



実施形態7のフローチャート

[Translation done.]

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開2001-312453

(P2001-312453A)

(43)公開日 平成13年11月9日(2001.11.9)

(51)Int.Cl. <sup>7</sup>	識別記号	F I	ターマコード*(参考)
G 0 6 F 13/10	3 2 0	G 0 6 F 13/10	3 2 0 A 5 B 0 1 4
9/06	4 1 0	9/06	4 1 0 C 5 B 0 7 6

審査請求 未請求 請求項の数15 O L (全 34 頁)

(21)出願番号 特願2000-128701(P2000-128701)

(22)出願日 平成12年4月28日(2000.4.28)

(71)出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72)発明者 山田 浩示

東京都大田区下丸子3丁目30番2号 キヤ

ノン株式会社内

(74)代理人 100075292

弁理士 加藤 卓

Fターム(参考) 5B014 EB01 FA11 FB04

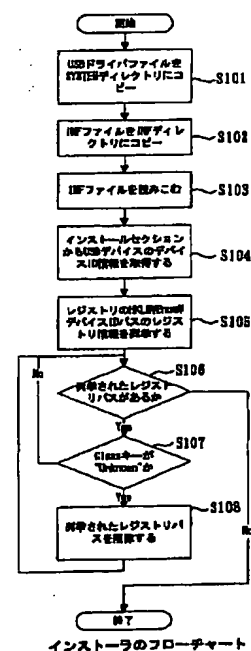
5B076 AA04 AA06 AA20

(54)【発明の名称】 電子機器、電子機器の制御方法、および電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体

(57)【要約】

【課題】 複数機器を接続可能なインタフェースに接続される周辺機器デバイスないしそのデバイスドライバの取り扱いを容易にする。

【解決手段】 USBドライバファイルをSYSTEMディレクトリにコピーし(S101)、INFファイルをINFディレクトリにコピーする(S102)。INFファイルをメモリに読み込み(S103)、INFファイル内のUSBのデバイスID情報を取得する(S104)。システムのレジストリを編集できるように読み込み(S105)、ステップS104で取得したデバイスIDを用いてレジストリパスを作り、このパス以下のレジストリ情報を列挙し、Classキーが“Unknown”になっている場合(S107)はそのレジストリパスの情報を全て削除する。“Unknown”でないなら次の列挙されたレジストリパスを調べるためステップS106へ戻る。



(図6)

## 【特許請求の範囲】

【請求項 1】 周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なうとともに、デバイスドライバの登録に関するレジストリ情報がレジストリ情報記憶手段に登録される電子機器において、

所定の周辺機器のデバイスドライバをシステムにインストールする際、登録すべきデバイスドライバに対応したデバイス ID 情報を前記レジストリ情報記憶手段で検索し、検索されたレジストリ情報の一部が不明となっていた場合、当該レジストリ情報を削除することを特徴とする電子機器。

【請求項 2】 所定の周辺機器のデバイスドライバをシステムにインストールする際、前記デバイス ID 情報がインストール情報定義ファイルから取り出されることを特徴とする請求項 1 に記載の電子機器。

【請求項 3】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続し、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なう電子機器において、

前記入出力ポートが増加したかどうか判断し、前記入出力ポートが増加した場合に増加したポートを介してそのポートに対応する周辺機器のデバイス名を取得し、取得したデバイス名が特定の対応機種を示している場合に当該周辺機器のデバイスドライバのインストール処理を行なうことを特徴とする電子機器。

【請求項 4】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器において、ユーザが入出力先として選択すべき周辺機器を確認するための確認コマンドを前記入出力ポート経由で送信し、該確認コマンドを受信した周辺機器側において該確認コマンドの受信をユーザに報知することを特徴とする電子機器。

【請求項 5】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器において、前記周辺機器の操作または動作状態に応じて定まるステータス情報を前記入出力ポートを介して前記周辺機器デバイスから受信し、受信したステータス情報に応じて入出力先として周辺機器を選択することを特徴とする電子機器。

【請求項 6】 周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なうとともに、デバイスドライバの登録に関するレジストリ情報がレジストリ情報記憶手段に登録される電子機器の制御方法において、所定の周辺機器のデバイスドライバをシステムにインストールする際、登録すべきデバイスドライバに対応した

デバイス ID 情報を前記レジストリ情報記憶手段で検索し、検索されたレジストリ情報の一部が不明となっていた場合、当該レジストリ情報を削除することを特徴とする電子機器の制御方法。

【請求項 7】 所定の周辺機器のデバイスドライバをシステムにインストールする際、前記デバイス ID 情報がインストール情報定義ファイルから取り出されることを特徴とする請求項 6 に記載の電子機器の制御方法。

【請求項 8】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続し、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なう電子機器の制御方法において、

前記入出力ポートが増加したかどうか判断し、前記入出力ポートが増加した場合に増加したポートを介してそのポートに対応する周辺機器のデバイス名を取得し、取得したデバイス名が特定の対応機種を示している場合に当該周辺機器のデバイスドライバのインストール処理を行なうことを特徴とする電子機器の制御方法。

【請求項 9】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器の制御方法において、ユーザが入出力先として選択すべき周辺機器を確認するための確認コマンドを前記入出力ポート経由で送信し、該確認コマンドを受信した周辺機器側において該確認コマンドの受信をユーザに報知することを特徴とする電子機器の制御方法。

【請求項 10】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器の制御方法において、前記周辺機器の操作または動作状態に応じて定まるステータス情報を前記入出力ポートを介して前記周辺機器デバイスから受信し、受信したステータス情報に応じて入出力先として周辺機器を選択することを特徴とする電子機器の制御方法。

【請求項 11】 周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なうとともに、デバイスドライバの登録に関するレジストリ情報がレジストリ情報記憶手段に登録される電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体において、

所定の周辺機器のデバイスドライバをシステムにインストールする際、登録すべきデバイスドライバに対応したデバイス ID 情報を前記レジストリ情報記憶手段で検索し、検索されたレジストリ情報の一部が不明となっていた場合、当該レジストリ情報を削除するための制御手順を格納したことを特徴とする電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体。

【請求項 12】 所定の周辺機器のデバイスドライバをシステムにインストールする際、前記デバイス ID 情報

がインストール情報定義ファイルから取り出されることを特徴とする請求項 1 に記載の電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体。

【請求項 13】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続し、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なう電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体において、

前記入出力ポートが増加したかどうか判断し、前記入出力ポートが増加した場合に増加したポートを介してそのポートに対応する周辺機器のデバイス名を取得し、取得したデバイス名が特定の対応機種を示している場合に当該周辺機器のデバイスドライバのインストール処理を行なうための制御手順を格納したことを特徴とする電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体。

【請求項 14】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体において、ユーザが入出力先として選択すべき周辺機器を確認するための確認コマンドを前記入出力ポート経由で送信し、該確認コマンドを受信した周辺機器側において該確認コマンドの受信をユーザに報知するための制御手順を格納したことを特徴とする電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体。

【請求項 15】 複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体において、前記周辺機器の操作または動作状態に応じて定まるステータス情報を前記入出力ポートを介して前記周辺機器デバイスから受信し、受信したステータス情報に応じて入出力先として周辺機器を選択するための制御手順を格納したことを特徴とする電子機器の制御プログラムを格納したコンピュータ読取可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続する電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体に関するものである。

【0002】

【従来の技術】 従来より、プリンタなどの周辺機器がコンピュータシステムとともに広く利用されている。従来では、プリンタはセントロニクスポートや RS232C のようなシリアルポートを介して、1 ポートに 1 台のみのプリンタをコンピュータ本体に接続することが多かつ

たが、近年になっていわゆる USB ポートのような、より汎用的なインタフェースを介して 1 ポートに複数台のプリンタを接続する形態が増えつつある。

【0003】

【発明が解決しようとする課題】 一方、マイクロソフト社の Windows (商品名: Windows 95、98、2000、NT などの異なるバージョンがある) ファミリーがパーソナルコンピュータの OS として広く用いられているが、USB デバイスのサポートを標榜しているものの、現在のところ標準で添付しているデバイスドライバはあまり多くない。

【0004】 たとえば、Windows 98 における USB ドライバのインストール手順は、他のドライバのインストールとほぼ同様、いわゆる SETUP. EXE のようなインストーラプログラムを手動で、あるいは「ウィザード」と呼ばれるプログラムにより、OS の起動時や、デバイスの PnP (プラグ・アンド・プレイ) 操作により自動ないし半自動的に起動することにより行なわれる。

【0005】 たとえば、OS 起動後に USB デバイスをパソコンに接続すると、OS は USB デバイスを検知し、ウィザードプログラムのデバイスドライバインストールダイアログを表示させる。このダイアログで、ユーザはデバイスドライバが格納されているディレクトリを正しく指定する必要がある。

【0006】 ユーザがダイアログを介してデバイスドライバが格納されているディレクトリ (CD-ROM や フロッピー (登録商標) ディスク上のディレクトリ) を指定すると、OS は指定されたディレクトリに検知した USB デバイス用のデバイスドライバが存在するか確認し、対応したものが存在すれば、これをインストールして終了する。

【0007】 ところが、ユーザが間違えたディレクトリを指定したため、検知した USB デバイスに対応していないデバイスドライバがあったり、そもそも当該のディレクトリにデバイスドライバが存在しない場合、OS は検知した USB デバイスを「不明なデバイス」として認識し、以後、そのデバイスは正常に機能しなくなる。この状態では、レジストリと呼ばれる OS の設定データに誤まったデータが登録されたままとなっており、このままでは再度 USB デバイスを繋ぎなおしても「不明なデバイス」として認識するため、ユーザはデバイスを全く使用できなくなる。

【0008】 このような状態になってしまった場合は、ユーザはデバイスマネージャなどのプログラムを用いて手動により当該デバイスを明示的に削除して正しい操作をやり直さなければ正常なデバイスドライバのインストールを行なうことができない。また、このような措置はある程度の経験を積んだユーザでなければ行なうのは困難であり、初心者は途方に暮れてしまう。また、この種



の些細なトラブルに関するユーザサポートに法外に高額な料金を要求するベンダーもあり、初心者であっても確実に失敗なくUSBデバイス（ドライバ）のインストールを行なえるようにすることが望まれているのはいうまでもない。

【0009】また、プリンタなどの機器をUSBポートに接続して用いる場合、この種のポートには複数の機器を同時に接続して用いることができるため、従来、存在しなかった新たな問題が発生しつつある。

【0010】すなわち、従来のプリンタはセントロニクスインタフェースなどのホストとデバイスの一対一通信しかできないインタフェースで接続されていたため、プリンタドライバ用のインストーラは例えばセントロニクスインタフェースからプリンタと直接通信を行い、プリンタ名などの情報を得て、適切なプリンタドライバをインストールしていた。

【0011】たとえば、Windowsの場合は通信インタフェースをポートと表現し、セントロニクスインタフェースのWindows上の名称（ポート名）はLPT1である。そこでデバイス（ドライバ）のインストーラはLPT1というポートをオープンし、このポートからプリンタに対してプリンタ名を要求し、取得したプリンタ名がインストーラの対応するプリンタ名と判断できたならインストール処理を行うようになっていた。

【0012】しかし近年普及しつつあるUSBインタフェースやIEEE1394インタフェースは一つのインタフェースに複数のデバイスが接続でき、しかもアドレスが動的に割り振られるため、あらかじめポート名を決めておくことができない。そのためあらかじめ決められたポート名を調べる従来の方法ではインストーラによる適切なプリンタドライバのインストールが行えず、ユーザに負担を強いることになっていた。

【0013】より具体的に問題点を挙げると次のようになる。Microsoft社のWindowsにおけるUSBインタフェースは、ホストとデバイスを接続しない限りポートと認識されない。しかもUSBインタフェースは最大127台のデバイスを接続可能であり、ポートと認識されたとしてもどのような名称になるか予測することができない。そのため従来のようにあらかじめ決められたポート名を使って接続機器、たとえばプリンタ（の機種）を検出することが困難になり、ユーザが適切なドライバを選択しなければならない、という問題があった。

【0014】あるいは、Windowsにおけるプラグアンドプレイ機能を用いることで、自動的にプリンタドライバをインストールすることもできるが、ベンダー独自のアプリケーションなどをインストールすることができないため独自の機能を追加することができない。そのためプリンタドライバのインストールにはインストーラが必要不可欠である。

【0015】以上の問題は、USBポート接続のプリンタに特別なものではなく、あらかじめデバイス名やアドレスなどを予め予測できない、という意味ではイーサネット（登録商標）などのネットワークを介して接続される機器においても同様にあてはまるものである。

【0016】また、外部機器がイーサネットなどのネットワークを介して接続される構成においては、同一機能の機器（たとえばプリンタ）が複数そのポートやネットワークに同時接続されることも稀ではなく、たとえば同一機種のプリンタが複数台そのポートやネットワークに接続されることもあり得る。

【0017】製品名とは別にニックネームのような名前を付けることでプリンタを区別する方法が一般的であった。しかしこの方法はニックネームを忘れると、どのプリンタが本当に印刷したいプリンタかわからなくなっていた。そのため試しに印刷してそれがどのプリンタから出てくるか確認しなければならなかった。そこで試し印刷をすることなく印刷したいプリンタを区別する方法が望まれている。

【0018】本発明の課題は、上記の問題を解決し、USBポートなどの複数機器を接続可能なインタフェースに接続されるデバイスドライバを誰でも間違いなくシステムにインストールでき、USBポートなどの複数機器を接続可能なインタフェースに接続されるプリンタなどの外部機器の識別を容易に行なうことができ、自分が使用したいプリンタなどの機器を容易に確認できるようにすることにある。

【0019】

【課題を解決するための手段】上記の課題を解決するため、本発明においては、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なうとともに、デバイスドライバの登録に関するレジストリ情報がレジストリ情報記憶手段に登録される電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、所定の周辺機器のデバイスドライバをシステムにインストールする際、登録すべきデバイスドライバに対応したデバイスID情報を前記レジストリ情報記憶手段で検索し、検索されたレジストリ情報の一部が不明となっていた場合、当該レジストリ情報を削除する構成を採用した。

【0020】あるいはさらに、所定の周辺機器のデバイスドライバをシステムにインストールする際、前記デバイスID情報がインストール情報定義ファイルから取り出される構成を採用した。

【0021】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続し、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納

したコンピュータ読取可能な記憶媒体において、前記入出力ポートが増加したかどうか判断し、前記入出力ポートが増加した場合に増加したポートを介してそのポートに対応する周辺機器のデバイス名を取得し、取得したデバイス名が特定の対応機種を示している場合に当該周辺機器のデバイスドライバのインストール処理を行なう構成を採用した。

【0022】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、ユーザが入出力先として選択すべき周辺機器を確認するための確認コマンドを前記入出力ポート経由で送信し、該確認コマンドを受信した周辺機器側において該確認コマンドの受信をユーザに報知する構成を採用した。

【0023】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、前記周辺機器の操作または動作状態に応じて定まるステータス情報を前記入出力ポートを介して前記周辺機器デバイスから受信し、受信したステータス情報に応じて入出力先として周辺機器を選択する構成を採用した。

【0024】

【発明の実施の形態】以下、図を参照して本発明の実施の形態を説明する。

【0025】〔第1の実施形態〕本実施形態では、Windows（商品名）上のデバイスドライバをインストールする処理を例示する。本実施形態では、デバイスドライバのインストールの際、いわゆるINFを参照してセットアップを実行するが、システムのレジストリを検索して対応するデバイスが「不明」なデバイスとなっている場合に自動的にレジストリから該当するレジストリパスを削除することにより、明示的な削除操作などを必要とすることなくデバイスドライバを誰でも間違いなくシステムにインストールできるようにする。

【0026】図1に本実施形態を実現するためのハードウェア構成を説明する。図示の構成は公知のパーソナルコンピュータの構成としてごく一般的なものである。

【0027】図1において符号H01は本実施形態の制御を司る制御部（以後CPUと称する）、H02は各種データを入力するキーボード、H03はRAMH04に貯えられている画像データを表示するカラー表示器で、CRTあるいはLCDディスプレイなどから構成される。

【0028】符号H04は装置全体を制御する制御手順およびその他の必要な情報をあらかじめ記憶するリードオンリーメモリ（以下ROMという）、H05はワーク

エリアとして利用されるランダムアクセスメモリ（以下RAMという）、H06は各種データの読み書きやプログラムが格納されているハードディスクである。後述のセットアッププログラムは、ハードディスクH06や不図示のCD-ROM、あるいはフロッピーディスクなどに格納された状態で供給され、RAMH05上（あるいはさらにハードディスクH06上に実装された仮想記憶機構を利用して）で実行される。

【0029】符号H07はUSBインタフェースであり、H08のUSBデバイスが接続される。USBデバイスH08としてはプリンタ、キーボード、マウス、タブレット、サウンドデバイス、デジタルカメラなど任意のデバイスが考えられる。H09はデータバスであり、アドレスバス、データバス、各種制御信号線などから構成され、各種データを転送するために用いられる。

【0030】図2に、本実施形態でインストールされるデバイスドライバのファイル構成を示す。図2のUSBDEV.SYSはUSBデバイス（この場合プリンタ）ドライバのファイルであり、動的にシステムに組み込むことができる一種の実行ファイルの形態で記述されたものである。

【0031】また、USBDEV.INFは上記のドライバをインストールするときに使われる設定ファイルである。その拡張子から、通称INFファイルと呼ばれている。SETUP.EXEは本発明を実施したインストールプログラムである。図2の各ファイルはハードディスクH06や不図示のCD-ROM、あるいはフロッピーディスクなどに格納された状態で供給される。たとえば、USB接続のプリンタを供給する場合、CD-ROM、あるいはフロッピーディスクなどに格納してそのプリンタとともにこれらのファイルが供給されるものとする。

【0032】なお、USBポートそのもののドライバは別途インストールされるものとする。

【0033】図3に上記のUSBDEV.INFの内容を示す。INFファイルは図示のように□で囲んだ名前によりセクションを定義し、その後に所定の制御パラメータの二重モニックとそのパラメータに定義すべき値を配列したテキストファイルであり、図中、左端の数字は行番号でありデータの一部をなすものではない。

【0034】1行目から4行目の[Version]セクションは、デバイスがUSBデバイスであり、デバイスのベンダー（Provider）は“Qanon”であることを示している。

【0035】続くセクション[DestinationDir]は、（5行目から6行目）はインストール先ディレクトリがWINDOWS¥SYSTEMディレクトリであることを示している。

【0036】[Manufacturer]セクション（7行目～8行目）は上記のProviderと下記の

Qanon. Sectionの関係を定義したものである。

【0037】[Qanon. Section] (9行目から10行目)はUSBのデバイスID“USB¥VID\_04A9&PID\_1052”のインストール処理を定義している。すなわち、デバイスID“USB¥VID\_04A9&PID\_1052”のインストールは、後述の[Qanon. USB]セクションの方法にしたがって行うことを定義している。また、ここでは名称は“Qanon USB Device”であることが定義されている。

【0038】[Qanon. USB]セクション(11行目から13行目)においては、インストール時のファイルコピーは後述の[Qanon. CopyList]セクションの情報に従うことを定義している。また、ここではレジストリ情報の追加は後述の[Qanon. AddReg]セクションの情報に従うことを示している。

【0039】[Qanon. CopyList]セクション(14行目から15行目)は、usbdev. sysファイル(図2)をインストールすることを示している。[Qanon. AddReg]セクション(16行目から23行目)は、システムのレジストリに追加すべき情報である。ここでは、ドライバの名称や、デバイスの種別、デバイスIDなどが定義されている。

【0040】次に図4を参照して、インストール時に“不明なデバイス”となった場合のレジストリの処理につき説明する。図4はレジストリエディタを用いて当該デバイスのエントリを開いた状態を示している。USBデバイスのレジストリ情報はHKEY\_LOCAL\_MACHINE¥Enum¥USB以下に記録される。レジストリは実際にはデータファイルの形式でハードディスクH06に記録されているが、レジストリエディタを用いてアクセスすると図示のような階層構造として表示される。「¥」はファイルシステムにおけるディレクトリのデリミタと同様に階層の境界を示す。

【0041】本デバイスの場合、インストール操作を行なうと、図4のようにHKEY\_LOCAL\_MACHINE¥Enum¥USB¥VID\_04A9&PID\_1052以下にデバイスが持つシリアル番号キーの中にデータが記録される。

【0042】図4はシリアル番号000000のデバイスを示しており、このレジストリ情報のClass情報がデバイスの種類を表す。図3のようなINFファイルを用いて正常にインストーラが実行された場合は、USBデバイスであればこのClass情報は“USB”となるが、インストールの際にユーザが間違ったディレクトリを指定したため、検知したUSBデバイスに対応していないデバイスドライバがあったり、そもそも当該のディレクトリにデバイスドライバが存在しない場合はこ

のClass情報の値は図4のように“不明なデバイス”を示す“Unknown”となる。

【0043】次に、図5に本実施形態のインストールプログラムのフローチャートを示す。図示のプログラムは、図2のSETUP. EXEの一部として実装され、CD-ROM、あるいはフロッピーディスクなどに格納して供給される。

【0044】図5のステップS101では、USBドライバファイルであるUSBDEV. SYSをWINDOWS¥SYSTEMディレクトリにコピーする。続いてステップS102ではINFファイルであるUSBDEV. INFをWINDOWS¥INFディレクトリにコピーする。

【0045】ステップS103ではUSBDEV. INFファイルをメモリに読み込み、ステップS104ではINFファイル内の[Qanon. Section]を探し、USBのデバイスID情報を取得する。ここで、図3のINFファイルの10行目の右辺の二つ目のパラメータがUSBのデバイスIDとなる。

【0046】ステップS105では、システムのレジストリを編集できるように読み込み、ステップS104で取得したデバイスIDの前にHKEY\_LOCAL\_MACHINE¥Enum¥を付加し、レジストリパスを作り、このパス以下のレジストリ情報を列挙する。図4のレジストリ内容であればデバイスID“USB¥VID\_04A9&PID\_1052”に相当するシリアル番号000000のデバイスが列挙される。

【0047】ステップS106～S108のループにおいて、同一デバイスIDのレジストリパスに複数のデバイスが登録されていれば、これらは順次列挙される。

【0048】ステップS106では列挙したレジストリパスが残っているか判断する。残っていればステップS107へ進み、残っていなければインストール処理を終了する。ステップS107ではClassキーが“Unknown”になっているか否かを判断する。“Unknown”ならステップS108へ進み、“Unknown”でないなら次の列挙されたレジストリパスを調べるためステップS106へ戻る。ステップS108では、Classキーが“Unknown”になっているレジストリパスの情報を全て削除する。

【0049】このようにして、本プログラムが対象とするデバイスドライバのインストールの際にユーザが間違ったディレクトリを指定したため、検知したUSBデバイスに対応していないデバイスドライバがあったり、そもそも当該のディレクトリにデバイスドライバが存在しない場合はこのClass情報の値は図4のように“不明なデバイス”を示す“Unknown”となっているレジストリパスの情報を全て削除することができるため、以後、USBデバイスを繋ぎ直すなどの操作を行ない、正しいディレクトリを指定するなどすればその時は

プリンタなどのUSBデバイスのドライバのインストールを正常に実行することができる。

【0050】したがって、ユーザは従来のようにレジストリエディタなどを用いて明示的ドライバを削除するような操作を必要とせず、また、高額なユーザサポートを必要とせずに容易に自分自身でUSBデバイスのインストールを行なうことができる。

【0051】本発明を実施したSETUP. EXEはフロッピーディスクやCD-ROMに限定されることなく任意のインストールメディアに格納して提供することができるのはいうまでもない。

【0052】なお、上記実施形態ではUSBデバイスのデバイスIDとしてUSB¥VID\_\_04A9&PID\_\_1052を用いたが、それ以外のデバイスIDでも適用できることは言うまでもない。また、以上ではUSBポートを例示したが、上記の技術は複数の外部機器を接続するインタフェース、たとえばIEEE1394のようなインタフェースであっても同様に実施できるのはいうまでもない。

【0053】[第2の実施形態] 以上では、インストーラ実行時のレジストリの操作につき示したが、以下では、USBなどのポートに動的に接続されるデバイスに適したインストーラの構成につき詳細に説明する。

【0054】本実施形態でも、Microsoft社のWindows上で動作するUSBプリンタに対応したインストーラである。USBプリンタの場合、2つのデバイスドライバを必要とする。1つはプリンタドライバ、もう1つはより低位のUSBドライバである。USBドライバはUSBでデータの送受信を行うためのデバイスドライバである。そのためUSBプリンタに対応したインストーラでは2つのデバイスドライバのインストールを行う必要がある。

【0055】本実施形態におけるインストール手順は次のようなものとなる。ユーザが本実施形態のインストーラを起動する（あるいはウィザードなどのプログラムにより自動起動される）と図7の画面が表示されるので、ユーザはUSBプリンタをホストと接続する。これにより後述のインストーラプログラムを介してUSBドライバがインストールされる。

【0056】USBポートそのもののドライバのインストール処理は前記実施形態同様に本実施形態のインストーラではなく、Windowsに付属のインストーラにより行なわれるものとする。また、USBドライバはインストールされると新しいポートを作成する。

【0057】以上の動作はWindowsにおけるUSBドライバの一般的な仕様である。一度USBドライバがインストールされると、本実施形態のインストーラは新しいポートを認識し、接続されたプリンタ名を得て最適なプリンタドライバを選択する。そしてファイルのコピー中は図8の画面を表示してファイルコピー処理を行

いプリンタドライバを登録する。ユーザはインストーラの起動とプリンタの接続だけ行えばよい。

【0058】本発明を実現するためのハードウェア構成を図6に示す。図6の構成は前記実施形態の図1とほぼ同様であるので、ここでは重複した説明は省略する。

【0059】図6において符号H07はプリンタインタフェースであり、H08はプリンタである。プリンタインタフェースH07としてはUSBが用いられるものとする。

【0060】次に図14と図15のフローチャートを用いて本実施形態のプリンタデバイスドライバのインストーラの動作を説明する。プリンタインストーラの実行ファイルの名称は前述の第1実施形態同様にSETUP. EXEであり、ハードディスクに格納された状態で、あるいは前述の実施形態同様にCD-ROMやフロッピーディスクなどのメディアにより供給される。

【0061】これらのメディア上のインストーラのバイナリ（SETUP. EXE）がユーザにより明示的に起動されるか、あるいはウィザード経由で起動されると、図14のステップS101において、USBドライバに関係するファイルをシステムにコピーする（図2参照）。USBドライバのファイルをあらかじめコピーすることで、ユーザはファイルを指定する必要がなくなる。

【0062】ステップS102はシステムが持つ全てのポート名を列挙し、ポートリストAとしてメモリに記憶するステップである。ポートリストの構造を図9に示す。図9はLPT1、COM1、FILE、およびUSBPRN01というポートを記憶した例である。

【0063】ステップS103では、図7のインストーラ画面を表示するステップで、この画面でユーザにUSBプリンタを繋ぐよう促す。ユーザがUSBポートにプリンタを繋ぐと、ステップS104においてポートチェックのインターバル時間になるまで待つ処理を入れる。本実施形態ではこのインターバル時間を2秒としている。2秒になったらステップS105へ進む。

【0064】ステップS105ではシステムが持つ全てのポート名を列挙し、ポートリストBとしてメモリに記憶する。

【0065】ステップS106ではポートリストBからポートリストAに存在するポートを削除する。この削除の結果、ポートリストBには新たに追加されたポートが残る。もしポートが追加されていなければ何も残らない。

【0066】ステップS107では、ポートリストBにポートが残っているか判断する。ポートが残っていなければステップS104へ戻り、ポートが残っていればステップS108へ進む。

【0067】ステップS108では、ポートリストBに残っているポート名がUSBインタフェースのポートか

否かを判断する。本実施形態ではUSBインタフェースのポート名は“USBPRN数字”となる。そこでポートリストBに残っているポート名の先頭が“USBPRN”かどうか判断し、先頭が“USBPRN”ならUSBインタフェースのポートだと判断する。USBインタフェースのポートだと判断したらステップS109へ進む。そうでなければステップS104へ戻る。

【0068】ステップS109ではUSBインタフェースのポートをオープンし、ステップS110ではオープンしたポートからUSBプロトコルのGET\_DEWC  
E\_IDリクエストを要求し、接続されたUSBデバイ  
10 スのデバイスIDを取得する。このデバイスIDは図10に示されるような文字列である。図10のデバイスIDは、デバイスの製造者が“Qan on”で製品名は“ABC-3000”ということを示している。

【0069】図15のステップS111では、ステップS109でオープンしたUSBインタフェースのポートをクローズする。

【0070】ステップS112はデバイスIDからプラグアンドプレイID（略してPnPID）を作るステップである。このPnPIDはデバイスIDのMFGタグの名称とMDLタグの名称を足したものである。図10のデバイスIDの場合、PnPIDは“Qan onABC-3000”となる。

【0071】ステップS113では、インストーラが持つ対応プリンタリストの中にステップS112で作成したPnPIDが記載されているか判断する。ここで、本実施形態の対応プリンタリストの構造を図11に示す。図11に示すように、対応プリンタリストはPnPIDと対応するファイル名から構成される。図11の例では、このインストーラは、“Qan on”社製の“ABC-300”、“ABC-80v”、“ABC-800α”の3機種に対応していることがわかる。もしステップS112で作成したPnPIDが対応プリンタリストに記載されていれば対応プリンタと判断しステップS114へ進み、記載されていなければステップS104へ戻る。

【0072】ステップS114では、ステップS103で表示したインストール画面1を消去し、ファイルのコピー中、ステップS115において図8のインストール画面2を表示する。ステップS116では、対応プリンタリストに記載されているファイル名を取得する。ステップS117では、ステップS116で取得したファイルをコピーする。

【0073】ステップS118では対応プリンタリストのファイルを全てコピーしたか判断する。全てのコピーが終了したらステップS119へ進む。コピーが終了していなければステップS117へ戻る。

【0074】ステップS119では、ステップS117でコピーしたファイルをプリンタドライバとしてシステ  
50 ムに登録する。ステップS120ではステップS115で表示したインストール画面2を消去する。

ムに登録する。ステップS120ではステップS115で表示したインストール画面2を消去する。

【0075】以上のようにして、入出力ポートが増えたかどうか判断するポート増加判定を行ない、増加したポートを介してプリンタ名を取得し、取得したプリンタ名がインストーラの対応するプリンタか判断し、対応機種と判断された場合インストール処理を行なうようにしている。USBインタフェースのような、ポート名があらかじめ決まっていないインタフェースでもプリンタのような機器を接続した際に最適なドライバをインストールすることができる。

【0076】【第3の実施形態】本実施形態は先の実施形態2より効率的にポートの追加を判断する例である。ハードウェア構成は上記実施形態と同様であるものとし、以下、図16と図17のフローチャートを用いて本実施形態におけるインストーラの動作を説明する。

【0077】プリンタインストーラの実行ファイルの名称は前述の第2実施形態同様にSETUP.EXEであり、ハードディスクに格納された状態で、あるいは前述の実施形態同様にCD-ROMやフロッピーディスクなどのメディアにより供給される。

【0078】これらのメディア上のインストーラのバイナリ（SETUP.EXE）がユーザにより明示的に起動されるか、あるいはウィザード経由で起動されると、図16のステップS201においてUSBドライバに関係するファイルをシステムにコピーする。USBドライバのファイルをあらかじめコピーすることで、ユーザはファイルを指定する必要がなくなる。

【0079】ステップS202では図7のインストーラ画面を表示する。この画面でユーザにUSBプリンタを繋ぐよう促す。ユーザがUSBプリンタを繋ぐと、ステップS203において、システムから送られるメッセージを待つ処理を行なう。ここで、メッセージとはシステムからアプリケーションに対して送られる命令である。例えば画面を書きなおすメッセージやキーが押された時にメッセージが送られる。

【0080】ステップS204では、システムから送られたメッセージがWM\_DEVICECHANGEメッセージかどうか判断する。このWM\_DEVICECHANGEメッセージは、パソコンと繋がっているデバイスの状態が変更されたときに送られる。ここでメッセージがWM\_DEVICECHANGEメッセージならステップS205へ進み、そうでなければステップS203へ戻る。

【0081】ステップS205では、WM\_DEVICECHANGEメッセージの要因を調べ、ポートが追加されたか否かを判断する。ポートが追加されたらステップS206へ進む。そうでなければステップS203へ戻る。

【0082】ステップS206では、追加されたポート

の名称の先頭が“USBPRN”かどうか判断する。ステップS206が肯定された場合にはステップS207へ進み、否定された場合にはステップS203へ戻る。ステップS207はステップS206でUSBと判断したポートをオープンするステップである。ステップS208はオープンしたポートからUSBプロトコルのGET\_DEVICE\_IDリクエストを要求し、デバイスIDを取得する。このデバイスIDは先の実施形態2と同じものである。ステップS209はステップS207でオープンしたUSBインタフェースのポートをクローズする。

【0083】ステップS210では、デバイスIDからプラグアンドプレイID(略してPnPID)を作る。前述の通り、PnPIDはデバイスIDのMFGタグの名称とMDLタグの名称を足したものである。

【0084】ステップS211では、インストーラが持つ対応プリンタリストの中にステップS210で作成したPnPIDが記載されているか否かを判断する。この対応プリンタリストは先の実施形態2と同じものである。ステップS210で作成したPnPIDが対応プリンタリストに記載されていれば対応プリンタと判断しステップS212へ進み、記載されていない場合はステップS203へ戻る。

【0085】ステップS212では、ステップS202で表示したインストール画面1を消去し、ステップS213では図8のインストール画面2を表示する。そして、ステップS214では対応プリンタリストに記載されているファイル名を取得する。さらに、ステップS215ではステップS214で取得したファイルをコピーするステップである。

【0086】ステップS216は対応プリンタリストのファイルを全てコピーしたか判断し、全てコピーしたらステップS217へ進む。ステップS217はステップS215でコピーしたファイルをプリンタドライバとしてシステムに登録する。さらに、ステップS218では、ステップS213で表示したインストール画面2を消去する。

【0087】以上のようにして、デバイス接続時のシステムから送られるメッセージが、WM\_DEVICECHANGEメッセージかどうか判断し、さらにその内容に応じて入出力ポートが増えたかどうか判断するポート増加判定を行ない、増加したポートを介してプリンタ名を取得し、取得したプリンタ名がインストーラの対応するプリンタか判断し、対応機種と判断された場合インストール処理を行なうようにしているので、USBインタフェースのような、ポート名があらかじめ決まっていな

率的にポートの追加を判断することができる。

【0088】[第4の実施形態]以下に、さらに異なる情報を用いて最適なプリンタを判断する例を示す。本実施形態で用いる情報はUSBデバイスが必ず持っている情報なので、プリンタ以外のデバイスのインストーラに応用することも可能である。ハードウェア構成は上記実施形態2および3と同様であるものとし、図18と図19のフローチャートを用いて本実施形態におけるインストーラの動作を説明する。

10 【0089】プリンタインストーラの実行ファイルの名称は前述の第3実施形態同様にSETUP.EXEであり、ハードディスクに格納された状態で、あるいは前述の実施形態同様にCD-ROMやフロッピーディスクなどのメディアにより供給される。

【0090】これらのメディア上のインストーラのバイナリ(SETUP.EXE)がユーザにより明示的に起動されるか、あるいはウィザード経由で起動されると、図18のステップS301に進む。

20 【0091】ステップS301ではUSBドライバに関係するファイルをシステムにコピーする。USBドライバのファイルをあらかじめコピーすることで、ユーザはファイルを指定する必要がなくなる。

【0092】ステップS302ではシステムが持つ全てのポート名を列挙し、ポートリストAとしてメモリに記憶する。ポートリストの構造は図9に示した通りで、ここでは図9はLPT1とCOM1とFILEとUSBPRN01というポートが記憶されている。

30 【0093】ステップS303は図7のインストーラ画面を表示するステップである。この画面でユーザにUSBプリンタを繋ぐよう促す。そして、ステップS304でポートチェックのインターバル時間になるまで待つ処理である。本実施形態では第2実施形態同様にインターバル時間を2秒とし、2秒になったらステップS305へ進む。

【0094】ステップS305ではシステムが持つ全てのポート名を列挙し、ポートリストBとしてメモリに記憶する。ステップS306では、ポートリストBからポートリストAに存在するポートを削除するステップである。削除した結果、ポートリストBには新たに追加されたポートが残る。もし追加されていなければ何も残らない。

【0095】ステップS307はポートリストBにポートが残っているか判断するステップである。ポートが残っていなければステップS304へ戻る。また、ポートが残っていればステップS308へ進む。

【0096】ステップS308はポートリストBに残っているポート名がUSBインタフェースのポートか判断するステップである。本実施形態ではUSBインタフェースのポート名は“USBPRN数字”となるので、ポートリストBに残っているポート名の先頭が“USBP

RN”かどうか判断し、先頭が“USBPRN”ならUSBインタフェースのポートであると判断する。USBインタフェースのポートの場合はステップS309へ進み、そうでなければステップS304へ戻る。

【0097】ステップS309はUSBインタフェースのポートをオープンする。ステップS310では、オープンしたポートからUSBプロトコルを用いてGET\_DESCRIPTORリクエストを要求し、デバイスデスクリプタを取得する。このデバイスデスクリプタはデバイスの基本的な情報を格納した構造体であり、USBデバイスに限らず全てのWindowsデバイスファイルに共通するデータである。デバイスデスクリプタの構造とその内容の例を図12に示す。図12において、デバイスデスクリプタのベンダーIDは製造者を意味するIDで、プロダクトIDは製品を区別するIDである。図12の場合ベンダーIDは04A9でプロダクトIDは0000となっている。

【0098】ステップS311では、ステップS309でオープンしたUSBインタフェースのポートをクローズする。そして、ステップS312では、ステップS310で取得したデバイスデスクリプタからベンダーIDとプロダクトIDを取り出す。

【0099】ステップS313では、インストーラが持つ対応プリンタリストの中にステップS312で取り出したベンダーIDとプロダクトIDが記載されているか判断する。本実施形態のインストーラが有する対応プリンタリストの構造を図13に示す。対応プリンタリストはベンダーIDとプロダクトIDと対応するファイル名から構成される。もしステップS312で作成したベンダーIDとプロダクトIDが対応プリンタリストに記載されていれば対応プリンタと判断しステップS314へ進み、記載がなければステップS304へ戻る。

【0100】ステップS314はステップS303で表示したインストール画面1を消去する。そして、ステップS315において図8のインストール画面2を表示する。インストール画面2はファイルのコピー中表示される。さらに、ステップS316は対応プリンタリストに記載されているファイル名を取得し、ステップS317はステップS316で取得したファイルをコピーする。

【0101】ステップS318は対応プリンタリストのプリンタのデバイスドライバのファイルを全てコピーしたか判断する。全てのコピーが終了したらステップS319へ進み、そうでなければステップS317に戻る。そして、ステップS319ではステップS317でコピーしたファイルをプリンタドライバとしてシステムに登録し、ステップS320ではステップS315で表示したインストール画面2を消去する。

【0102】以上のようにして、GET\_DESCRIPTORプロトコルを用いてデバイスデスクリプタを取得し、デバイス名(プリンタ名)を取得し、入出力ポー

トが増えたかどうか判断するポート増加判定を行ない、増加したポートを介してプリンタ名を取得し、取得したプリンタ名がインストーラの対応するプリンタか判断し、対応機種と判断された場合インストール処理を行なうようにしているので、USBインタフェースのような、ポート名があらかじめ決まっていないインタフェースでもプリンタのような機器を接続した際に最適なドライバをインストールすることができる。

【0103】本実施形態はUSBインタフェースに限定したものであるが、デバイスドライバがインストールされるとポートが追加されるインタフェースであれば、どのようなインタフェースを用いても実施できる利点がある。

【0104】以上では、OSとしてMicrosoft社のWindowsを考えたが、任意のOSにおいてプリンタなどのデバイスを特定するのに類似のメカニズムを用いている場合は、上記の技術を応用できるのはいうまでもない。

【0105】[第5の実施形態] 本実施形態では、USBポート、イーサネットなどのインタフェースを介して複数のプリンタなどの外部機器を接続する場合に、自分が使用したい外部機器を容易に確認できるようにするための構成を例示する。

【0106】本実施形態では、ホストで選んだプリンタに「確認コマンド」を送信し、それにプリンタを応答させることにより自分が使用したいプリンタを容易に確認できるようにする。より具体的には、「確認コマンド」を受けたプリンタは可聴音を発生し、これにより、ユーザがどのプリンタから音が鳴ったか判断する構成である。これにより、ホストで表示しているプリンタと実際のプリンタとの対応付けを確認することができる。

【0107】最初に本実施形態を実現するためのハードウェア構成を図20を用いて説明する。図20は、図1あるいは図6と同等の図面で、同一ないし相当する部材には同一符号を付してある。プリンタインタフェースH07はここでもUSBポートのような複数台のプリンタH81、H82…を同時接続できるようなポートであるものとする。それ以外の構成は図1あるいは図6と同じであるので、ここでは詳細な説明を省略する。

【0108】本実施形態においては、コンピュータ(ホスト)からプリンタH81、H82…に「確認コマンド」を送信するが、その際に図21のフローチャートのような制御手順と、図22のようなダイアログを用いる。

【0109】図21の制御手順は、CPU H01が実行可能なプログラムとしてハードディスクに格納された状態で、あるいは前述の実施形態同様にCD-ROMやフロッピーディスクなどのメディアにより供給される。より具体的には、このようなプログラムはプリンタドライバプログラムの一部として供給し、コントロールパネ

ルやプリンタ制御用のメニューやアイコンから、あるいはアプリケーションのプリントダイアログ経由で起動されるものとする。メニューやアイコンのタイトルは「プリンタ確認」「プリンタ選択」などのものが考えられる。

【0110】上記のようなメニューやアイコンにより起動されると、同プログラムは、図21のステップS01において、ホストと接続しているプリンタ名をリスト表示する。図22の真ん中左側がホストと接続しているプリンタ名をリストするための表示領域で、プリンタ数が多ければこの領域はスクロール可能に制御される。プリンタ名は、図12に示したようなGET\_DESCRIPTORプロトコルを用いて取得したプロダクト名文字列を用いることができる他、同一のプリンタが存在する場合にはさらにこのプロダクト名文字列の後に自動的に番号を付与して生成した文字列を用いたり、あるいはユーザが任意に設定したニックネームを用いることも考えられる。

【0111】ステップS02では、ユーザからのマウスクリック待ちを行なう。ここでは、ユーザは所望のプリンタ名をマウスでクリックすることにより、選択し、「確認」ボタンを押下することにより確認コマンドの送信を行なう。また、「選択」ボタンを押下することにより印刷に用いるプリンタを選択することができる。

【0112】クリックイベントが発生した場合には、ステップS03において、図22のダイアログのプリンタ名の部分がクリックされたか否かを判断する。プリンタ名がクリックされると、ステップS04とステップS05が実行される。すなわち、ステップS04は今まで反転表示していたプリンタ名を元に戻し、ステップS05では新たにクリックされたプリンタ名を反転表示する。

【0113】ステップS06、S08、S10はそれぞれ図22のダイアログの「確認」、「キャンセル」、「選択」の各ボタンのクリックを検出するものである。

【0114】まず、ステップS06では、図22の「確認」ボタンが押されたか否かを判断し、「確認」ボタンが押されると、ステップS07において、現在反転表示中のプリンタに対して確認コマンドを送る。この確認コマンドのフォーマットは任意であり、本実施形態ではプリンタのスピーカなどを用いて可聴音を発生させることができる。たとえばそのプリンタのPDL（ページ記述言語）などに対応するコマンド（警告音発生のためのコマンドなど）があればそれを用いることができる。プリンタはこのコマンドを受信するとピープ音を鳴らす。

【0115】ステップS08で図22の「キャンセル」ボタンが押された場合は、ステップS09において全てのプリンタ選択を無効としダイアログの処理を終了する。ステップS10で「選択」ボタンが押された場合は、ステップS11で現在反転表示しているプリンタを

新たに選択したプリンタとしてダイアログの処理を終了する。ステップS10で「選択」ボタンが押されていない場合はステップS02へ戻り新たなマウスクリックを待つ。

【0116】一方、プリンタ側では、図23のフローチャートのような処理を行なう。すなわち、ステップS01ではコマンドの受信待ちを行ない、コマンドを受信した場合はステップS02で受信したコマンドが図21のステップS07で送信された確認コマンドかどうかを判断し、確認コマンドだった場合、ステップS03でピープ音を鳴らす。

【0117】ステップS04以後は本来のプリンタの処理であり、受信コマンドが印刷コマンドであれば対応する印刷処理を行なう（ステップS05）。

【0118】このようにして、ユーザはダイアログ（図22）を用いてプリンタのリストを調べ、目的のプリンタに確認コマンドを送信してピープ音を発生させることができるので、自分が使用したいプリンタを容易に確認することができる。

【0119】上記の技術は、プリンタに限らず、USBポート、イーサネットなどのインタフェースを介して複数の外部機器（たとえばスキャナなどの入力機器であってもよい）を接続する場合に、自分が使用したい外部機器を容易に確認できるようにするために用いることができるものである。

【0120】なお、本実施形態では確認コマンドとして音を鳴らすコマンドを用いたが、本発明の本質は印刷と異なる動作をするコマンドをプリンタに実行させることで、ユーザが印刷先のプリンタを確認できるように報知することであるから、確認コマンドに応じてプリンタで実行させる動作は可聴音の発生だけではなく、例えばLEDを点滅させるような動作であってもよい。

【0121】[第6の実施形態] 上記の第5実施形態とは逆に、外部機器（プリンタ）側で操作を行なうことにより、その外部機器を出力（あるいは入力）に用いるべく「選択」状態とすることも考えられる。

【0122】たとえば、図26に示すように、外部機器としてのプリンタ（図20のH81あるいはH82に相当する）に選択ボタンB02を設けておき、ユーザがそのプリンタを出力に用いたい場合、この選択ボタンB02を押下する。この選択ボタンB02の押下は所定のプロトコルにより、ホスト側に伝達され、これを受信したホスト側ではそのプリンタを出力に用いるべく「選択」状態に制御する。なお、図26のボタンB01は電源ボタンである。

【0123】本実施形態のホスト（コンピュータ）側のハードウェア構成は第5実施形態と同一でよい。本実施形態においては、図24のフローチャートのような制御手順と図25のようなダイアログを用いる。図24のプログラムは、第5実施形態と同様の契機で起動すればよ



い。図24の処理の全体構成は図21とほぼ同様であるので、以下では図21の処理と同様の細部については説明を省略する。

【0124】図24のステップS01では、現在接続しているプリンタを図25のダイアログに表示する。

【0125】ステップS02ではマウス入力待ち、マウス入力があった場合はまずステップS03でステップS01で表示した全プリンタのステータスを得る。このステータスには図26の選択ボタンB02が押されているか否かを示す「選択ステータス」が含まれているものとする。前述のように、この選択ボタンB02の押下は所定のプロトコルによりホスト側に伝達される。この選択ステータスが1なら当該のプリンタで選択ボタンB02が押されたことを意味している。

【0126】ステップS04では「選択ステータス=1」のステータスを返したプリンタがあるかを判断し、選択ステータス=1のプリンタがあればステップS05へ進み、今まで反転表示していたプリンタ名の表示を元に戻し、さらにステップS06で選択ステータス=1となっていたプリンタ名の表示を反転する。

【0127】ステップS07はステップS02で得たマウス入力情報からマウスクリックが発生したか判断し、マウスクリックが発生したらステップS08へ進み、マウスクリックが発生していなければステップS02へ戻る。

【0128】ステップS08は図25のダイアログのプリンタ名がクリックされたか判断し、クリックされていればステップS09へ進み、クリックされていなければステップS11へ進む。

【0129】ステップS09では、今まで反転表示していたプリンタ名の表示を元に戻し、ステップS10ではマウスクリックされたプリンタ名の表示を反転するステップである。すなわち、ここでは図21同様に手動操作によりプリンタの選択を行なえるようにしている。

【0130】ステップS11では、図25のダイアログの「キャンセル」ボタンが押されたかを判断する。同ボタンが押されていればステップS12へ進み、プリンタの選択を全て無効とし、ダイアログの処理を終了する。

【0131】ステップS13では、図25の「選択」ボタンが押されたか否かを判断し、押されていなければステップS02へ戻り新たなマウス入力を受け付ける。一方、確認ボタンが押された場合はステップS14において現在反転表示しているプリンタを新たに選択したプリンタとしてダイアログの処理を終了する。

【0132】プリンタ側では、図27のフローチャートのような処理を行なうことにより選択ステータスを決定することができる。図27は選択ステータスに関する処理を行うタスクを説明するものであり、プリンタ本来の処理は別のタスクとして動作させればよいので、ここではプリンタ本来の処理については説明を省略する。

【0133】図27のステップS01はプリンタの操作ボタンの押下を受け付け、ステップS02では選択ボタン(図26のB02)が押されたか判断する。選択ボタンが押された場合ステップS03へ進み、選択ステータスを1に設定し、選択ボタンが押されていない場合にはステップS04において選択ステータスを0に設定する。プリンタはこのステータスを保持し、ステータス要求があったときに所定のプロトコルを用いてホストへ送信すればよい。

10 【0134】このようにして、ユーザはダイアログ(図25)を用いてプリンタのリストを調べた時、プリンタ側で選択ボタンを押下していれば既にそのプリンタが反転表示状態となっているので、単にその選択状態を「選択」ボタンにより追認するだけで自分が使用したいプリンタを容易に確認し、選択することができる。

【0135】本実施形態の技術も、プリンタに限らず、USBポート、イーサネットなどのインタフェースを介して複数の外部機器(たとえばスキャナなどの入力機器であってもよい)を接続する場合に、自分が使用したい外部機器を容易に確認できるようにするために用いることができるものである。

20 【0136】本実施形態では、専用の「選択」ボタンをプリンタに設ける構成を例示したが、例えばオンラインボタンや他の既存の操作ボタン、あるいは複数の既存の操作ボタンの押下の組合せを「選択」操作に用いるようにしてもかまわない。

【0137】[第7の実施形態]あるいは、本実施形態はプリンタの電源を入れるとホストのプリンタドライバで自動的にそのプリンタを印刷先として「選択」させることも考えられる。本実施形態のハードウェア構成は実施形態1と同じでよい。

【0138】本実施形態においては、図28のフローチャートのような制御手順と図25のダイアログ(第6実施形態と同じ)を用いる。図28のプログラムは、第5、6実施形態と同様の契機で起動すればよい。図28の処理の全体構成は図21、24とほぼ同様であるので、以下では図21、24の処理と同様の細部については説明を省略する。

【0139】図28のステップS01では現在接続しているプリンタを図25のダイアログで表示し、ステップS02でマウス入力待つ。

40 【0140】ステップS03では新たに接続されたプリンタがあるかどうか判断する。ここで、「接続」とはUSBバスや、イーサネットなどのネットワーク上でプリンタが見えるようになればよい。そのためUSBバスや、イーサネットなどを介して接続されたプリンタが電源オフの状態から電源オンの状態に変わったときもこのステップで検出できる。具体的には、バスあるいはネットワーク上のプリンタのアドレスを監視したり、図26の電源ボタンB01などにより電源が投入された時にプ

リント側から明示的な信号を送信するようにしても、新たに「接続」されたプリンタがあるかどうかを判断することができる。

【0141】新たに接続されたプリンタがあった場合はステップS04において、今までダイアログ上で反転表示していたプリンタ名の表示を元に戻し、ステップS05で新たに「接続」されたプリンタのプリンタ名の表示を反転する。

【0142】ステップS06では、ステップS02で得たマウス入力情報からマウスクリックが発生したか判断し、マウスクリックが発生したらステップS07へ進み、発生していなければステップS02へ戻る。

【0143】ステップS07では図25のダイアログのプリンタ名をクリックされたかを判断し、クリックされていればステップS08へ進み、クリックされていなければステップS10へ進む。

【0144】ステップS08では今まで反転表示していたプリンタ名の表示を元に戻し、ステップS09ではマウスクリックされたプリンタ名の表示を反転する。

【0145】ステップS10は図25のダイアログの「キャンセル」ボタンが押されたかを判断し、押されていればステップS11へ進む。ステップS11ではプリンタの選択を全て無効とし、ダイアログの処理を終了する。

【0146】ステップS12では図25のダイアログの「選択」ボタンが押されたかどうかを判断し、押されていなければステップS02へ戻り新たなマウス入力待つ。確認ボタンが押された場合にはステップS13において現在反転表示しているプリンタを新たに選択したプリンタとしてダイアログの処理を終了する。

【0147】このようにして、ユーザはダイアログ(図25)を用いてプリンタのリストを調べた時、電源投入したプリンタが反転表示状態となっているので、単にその選択状態を「選択」ボタンにより追認するだけで自分が使用したいプリンタを容易に確認し、選択することができる。

【0148】本実施形態の技術も、プリンタに限らず、USBポート、イーサネットなどのインタフェースを介して複数の外部機器(たとえばスキャナなどの入力機器であってもよい)を接続する場合に、自分が使用したい外部機器を容易に確認できるようにするために用いることができるものである。

【0149】

【発明の効果】以上の説明から明らかなように、本発明によれば、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なうとともに、デバイスドライバの登録に関するレジストリ情報がレジストリ情報記憶手段に登録される電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、所定

の周辺機器のデバイスドライバをシステムにインストールする際、登録すべきデバイスドライバに対応したデバイスID情報を前記レジストリ情報記憶手段で検索し、検索されたレジストリ情報の一部が不明となっていた場合、当該レジストリ情報を削除する構成を採用しているため、明示的ドライバを削除するような操作を必要とせず、また、高額なユーザサポートを必要とせずに容易に自分自身でUSBデバイスのインストールを行なうことができる、という優れた効果がある。

【0150】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続し、周辺機器デバイスを当該機器のデバイスドライバ経由で制御することにより当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、前記入出力ポートが増加したかどうか判断し、前記入出力ポートが増加した場合に増加したポートを介してそのポートに対応する周辺機器のデバイス名を取得し、取得したデバイス名が特定の対応機種を示している場合に当該周辺機器のデバイスドライバのインストール処理を行なう構成を採用することにより、USBインタフェースのように、ポート名があらかじめ決まっておらず、複数の周辺機器を同時接続可能な入出力ポートでも、プリンタのような機器を接続した際に最適なドライバをインストールすることができる、という優れた効果がある。

【0151】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、ユーザが入出力先として選択すべき周辺機器を確認するための確認コマンドを前記入出力ポート経由で送信し、該確認コマンドを受信した周辺機器側において該確認コマンドの受信をユーザに報知する構成を採用することにより、USBインタフェースのように、ポート名があらかじめ決まっておらず、複数の周辺機器を同時接続可能な入出力ポートでも、ユーザは自分の使用すべきプリンタなどの周辺機器を容易に特定することができる、という優れた効果がある。

【0152】あるいは、複数の周辺機器を同時接続可能な入出力ポートを介して周辺機器デバイスを接続して当該機器に対する入出力を行なう電子機器、その制御方法、およびその制御プログラムを格納したコンピュータ読取可能な記憶媒体において、前記周辺機器の操作または動作状態に応じて定まるステータス情報を前記入出力ポートを介して前記周辺機器デバイスから受信し、受信したステータス情報に応じて入出力先として周辺機器を選択する構成を採用することにより、周辺機器の操作または動作状態、たとえば、周辺機器の操作手段の操作状態や主電源の投入状態などに応じて定まるステータスを用いて自分が使用したいプリンタなどの周辺機器を容易

に確認し、選択することができる、という優れた効果が得られる。

【図面の簡単な説明】

【図1】本発明を実施可能なパーソナルコンピュータのハードウェア構成を示したブロック図である。

【図2】本発明によるデバイスドライバのインストール処理に用いられるファイルの構成を示した説明図である。

【図3】図2のUSBDEV.INFの内容を示した説明図である。

【図4】レジストリエディタによりシステムのレジストリをオープンした状態を示した説明図である。

【図5】本発明の第1実施形態における制御手順を示したフローチャート図である。

【図6】本発明を実施可能なパーソナルコンピュータのハードウェア構成を示したブロック図である。

【図7】本発明において表示されるインストーラ画面1を示した説明図である。

【図8】本発明において表示されるインストーラ画面2を示した説明図である。

【図9】本発明において用いられるポートリストを示した説明図である。

【図10】本発明において用いられるデバイスIDを示した説明図である。

【図11】本発明のインストーラプログラムが有する対応プリンタリストを示した説明図である。

【図12】本発明において用いられるデバイスディスクリプタを示した説明図である。

【図13】本発明のインストーラプログラムの異なる対応プリンタリストを示した説明図である。

【図14】本発明の第2実施形態における制御手順を示したフローチャート図である。

【図15】本発明の第2実施形態における制御手順を示したフローチャート図である。

【図16】本発明の第3実施形態における制御手順を示\*

\*したフローチャート図である。

【図17】本発明の第3実施形態における制御手順を示したフローチャート図である。

【図18】本発明の第4実施形態における制御手順を示したフローチャート図である。

【図19】本発明の第4実施形態における制御手順を示したフローチャート図である。

【図20】本発明を実施可能なパーソナルコンピュータのハードウェア構成を示したブロック図である。

10 【図21】本発明の第5実施形態における制御手順を示したフローチャート図である。

【図22】本発明において用いられるプリンタ選択ダイアログを示した説明図である。

【図23】プリンタ側の制御を示したフローチャート図である。

【図24】本発明の第6実施形態における制御手順を示したフローチャート図である。

【図25】本発明において用いられるプリンタ選択ダイアログを示した説明図である。

20 【図26】選択ボタンを設けたプリンタの構成を示した説明図である。

【図27】プリンタ選択ステータスを示したフローチャート図である。

【図28】本発明の第7実施形態における制御手順を示したフローチャート図である。

【符号の説明】

H01 CPU  
H02 キーボード  
H03 表示器  
H04 ROM  
H05 RAM  
H06 ハードディスク  
H07 USBインタフェース  
H08 USBデバイス

【図2】

(図2)

ファイル名	説明
USBDEV.SYS	デバイスドライバ本体
USBDEV.INF	設定ファイル
SETUP.EXE	インストーラプログラム

ファイル構成図

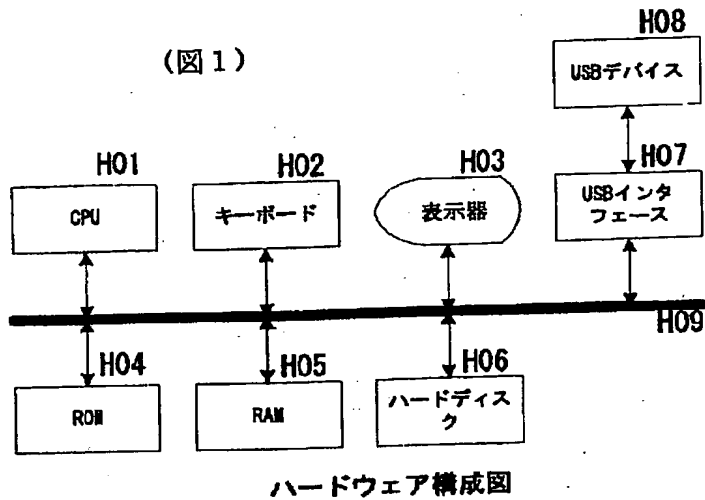
【図9】

(図9)

番号	ポート名
1	LPT1
2	COM1
3	FILE
4	USBPRN01

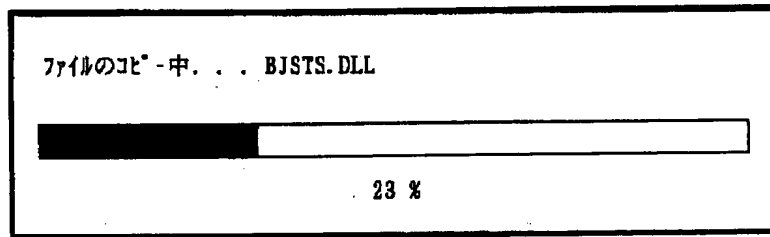
ポートリスト

【図1】



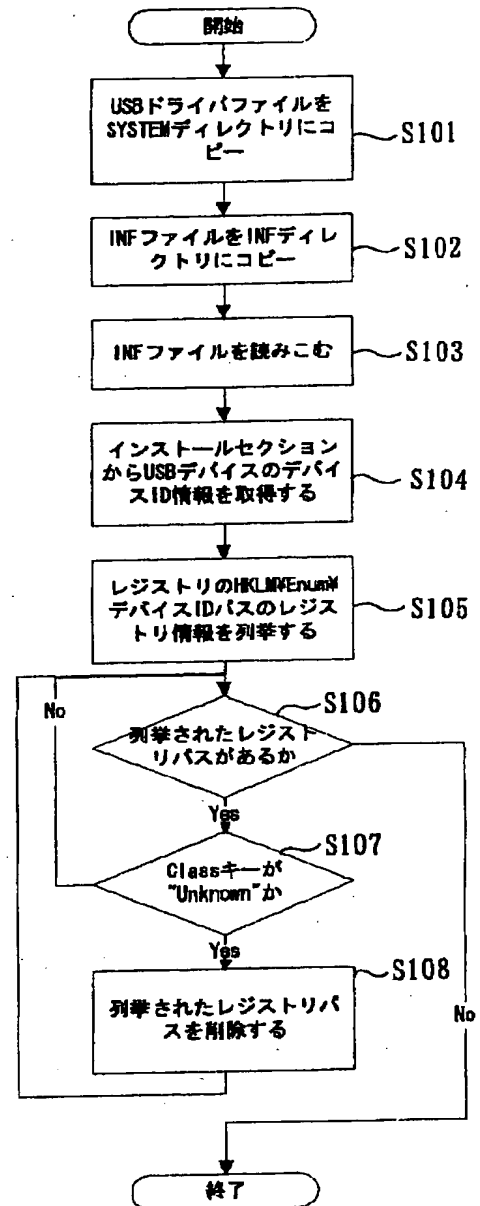
【図8】

(図8)



インストーラ画面2

【図5】



インストーラのフローチャート

(図5)

【図10】

(図10)

MFG:Qanon;CMD:TEXT;MDL:ABC-3000;CLS:PRINTER;

デバイスID

【図3】

(図3)

```
1. [Version]
2. Signature="$Windows 95$"
3. Class=USB
4. Provider=Qanon

5. [DestinationDirs]
6. DefaultDestDir=11

7. [Manufacturer]
8. Qanon=Qanon.Section

9. [Qanon.Section]
10. "Qanon USB Device"=Qanon.USB,USB\VID_04A9&PID_1052

11. [Qanon.USB]
12. CopyFiles=Qanon.CopyList
13. AddReg=Qanon.AddReg

14. [Qanon.CopyList]
15. usbdev.sys

16. [Qanon.AddReg]
17. HKR,,DevLoader,,*ntkern
18. HKR,,NTMPDriver,,usbdev.sys
19. HKR,,QanonSoft,,Printer
20. HKR,,Device,,USBPRN
21. HKR,,ID,,VID_04A9&PID_1052
22. HKR,,Instance,,Multiple
23. HKR,,DeviceDesc,, "Qanon USB Device"

24. [ControlFlags]
25. ExcludeFromSelect=USB\VID_04A9&PID_1052

26. [PreCopySection]
27. HKR,,NoSetupUI,,1

28. [SourceDisksNames]
29. 1 = "Qanon USB Device Disk",,0001

30. [SourceDisksFiles]
31. usbdev.sys = 1
```

USBDEV. INFの内容

【図4】

レジストリ エディタ	
名前	データ
<input type="checkbox"/> Enum	
<input type="checkbox"/> BIOS	
<input type="checkbox"/> ESDI	
<input type="checkbox"/> FLOP	
<input type="checkbox"/> HTREE	
<input type="checkbox"/> INFRARED	
<input type="checkbox"/> ISAPNP	
<input type="checkbox"/> LPTENUM	
<input type="checkbox"/> MF	
<input type="checkbox"/> MONITOR	
<input type="checkbox"/> Network	
<input type="checkbox"/> PCI	
<input type="checkbox"/> Root	
<input type="checkbox"/> SCSI	
<input type="checkbox"/> USB	
<input type="checkbox"/> ROOT_HUB	
<input type="checkbox"/> VID_047E&PID_1001	
<input type="checkbox"/> VID_04A9&PID_1052	
<input checked="" type="checkbox"/> 55C330	

名前	データ
<input type="checkbox"/> (標準)	(値の設定なし)
<input type="checkbox"/> Capabilities	14 00 00 00
<input type="checkbox"/> Class	"Unknown"
<input type="checkbox"/> CompatibleIds	"USB\CLASS_07&SUBCLASS_01&PROT_02.USB\CLA
<input type="checkbox"/> ConfigFlags	40 00 00 00
<input type="checkbox"/> DeviceDesc	"Qanon USB Device"
<input type="checkbox"/> HardwareId	"USB\VID_04A9&PID_1052&REV_0103.USB\VID_0

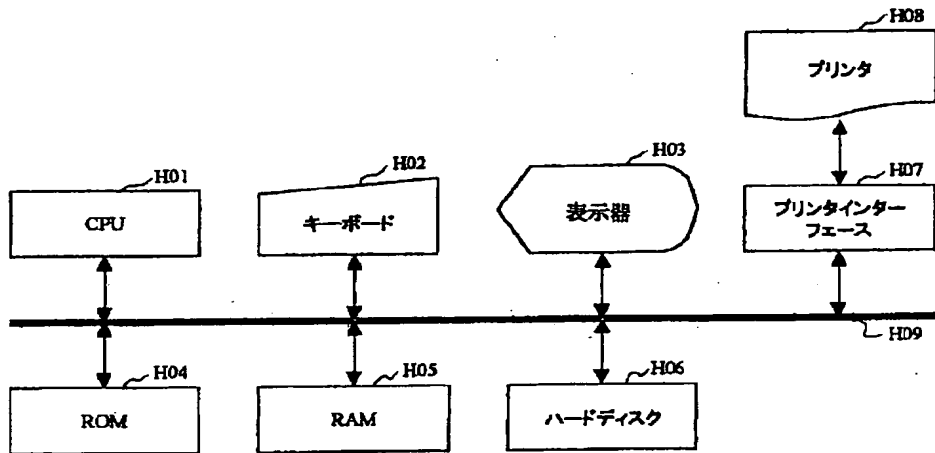
マイコンビュータ\FKBY LOCAL MACHINE\Enum\USB\VID\_04A9&PID\_1052\F0000000

レジストリ

(図4)

【図6】

(図6)



ハードウェアブロック図

【図11】

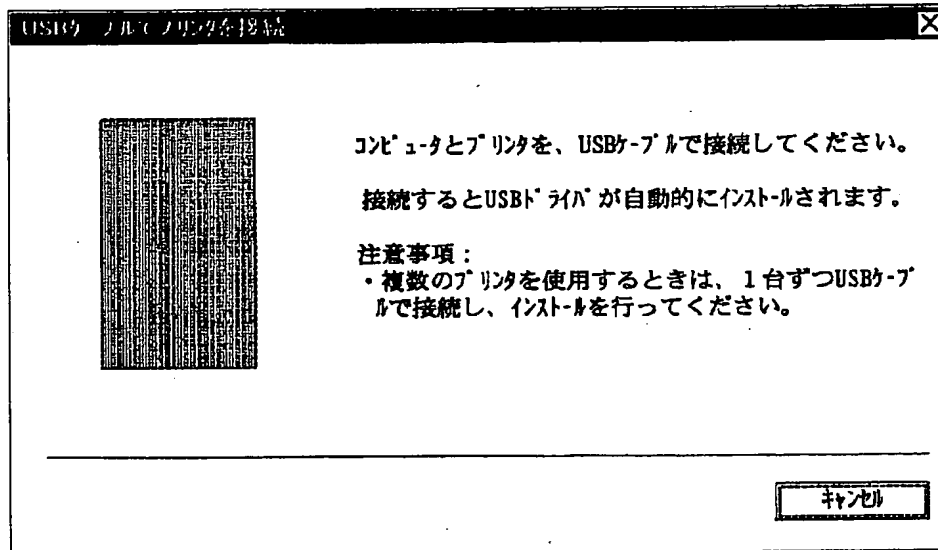
(図11)

番号	PnP ID	必要ファイル
1	QanonABC-3000	ABC3000.DRV ABC3000.DLL QANON_UI.DLL QANON_UN.EXE
2	QanonABC-80v	ABC80V.DRV ABC80V.DLL QANON_UI.DLL QANON_UN.EXE
3	QanonABC-8000	ABC8000.DRV ABC8000.DLL QANON_UI.DLL QANON_UN.EXE

対応プリンタリスト

【図7】

(図7)



インストーラ画面1

【図12】

(図12)

フィールド名	値(16進数)	意味
bLength	18	デバイスデスク립タのバイトサイズ
bDescriptorType	01	デバイスデスク립タのタイプ番号
bcdUSB	0100	USBバージョン番号
bDeviceClass	00	デバイスクラスコード
bDeviceSubClass	00	デバイスサブクラスコード
bDeviceProtocol	00	デバイスプロトコルコード
bMaxPacketSize	08	最大パケットサイズ
idVender	04A9	ベンダーID
idProduct	0000	プロダクトID
bcdDevice	0100	デバイスバージョン番号
iManufacturer	01	ベンダー名文字列番号
iProduct	02	プロダクト名文字列番号
iSerialNumber	03	シリアル番号文字列番号
bNumConfigurations	01	コンフィグレーション数

デバイスデスク립タの構造



【図13】

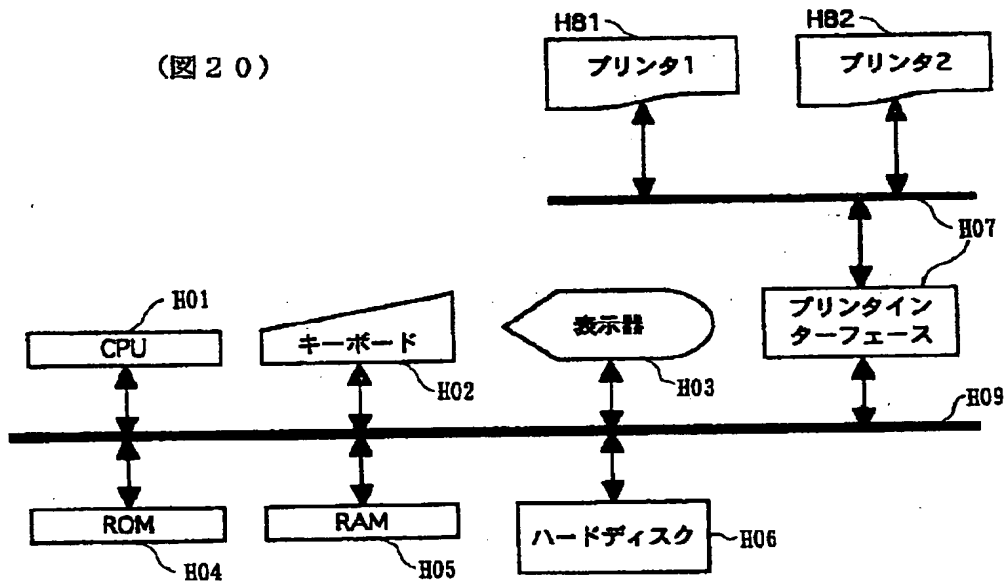
(図13)

番号	ベンダーID	プロダクトID	必要ファイル
1	04A9	0001	ABC3000.DRV ABC3000.DLL QANON_UI.DLL QANON_UN.EXE
2	04A9	0002	ABC80V.DRV ABC80V.DLL QANON_UI.DLL QANON_UN.EXE
3	04A9	0003	ABC8000.DRV ABC8000.DLL QANON_UI.DLL QANON_UN.EXE

実施形態3の対応プリンタリスト

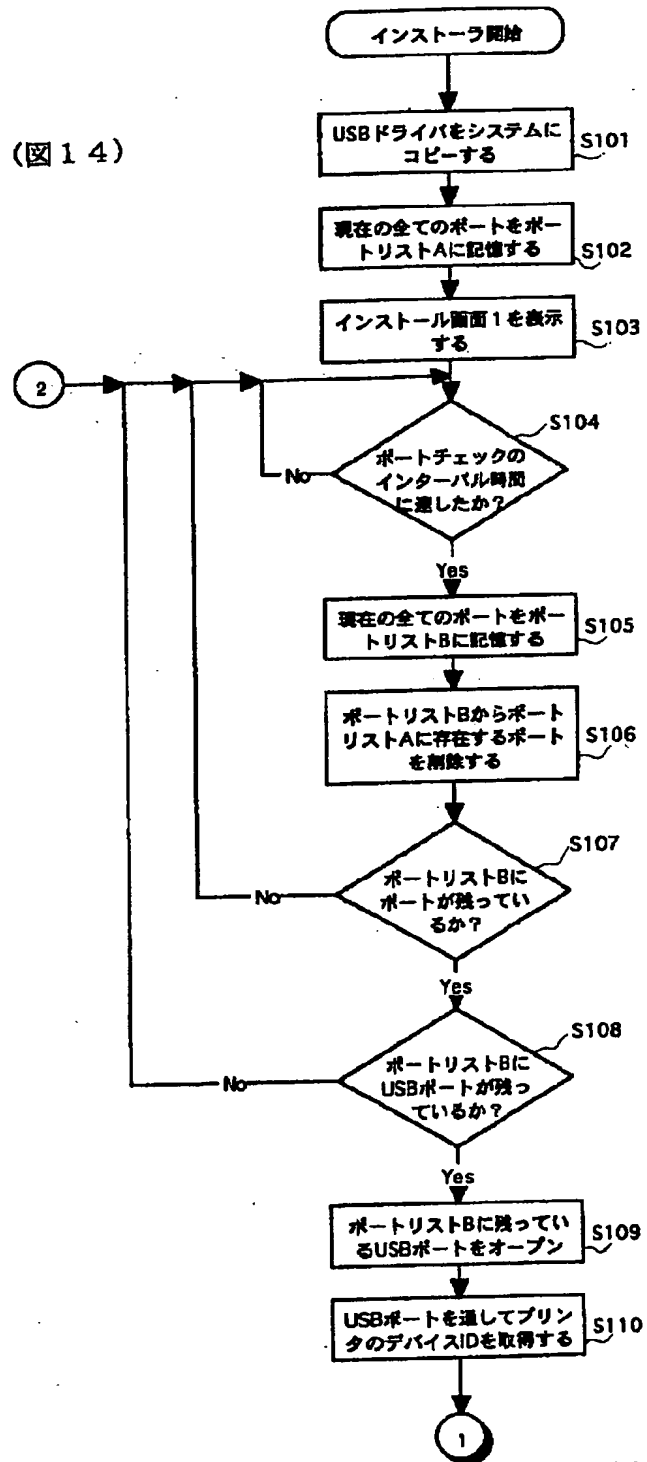
【図20】

(図20)



ハードウェアブロック図

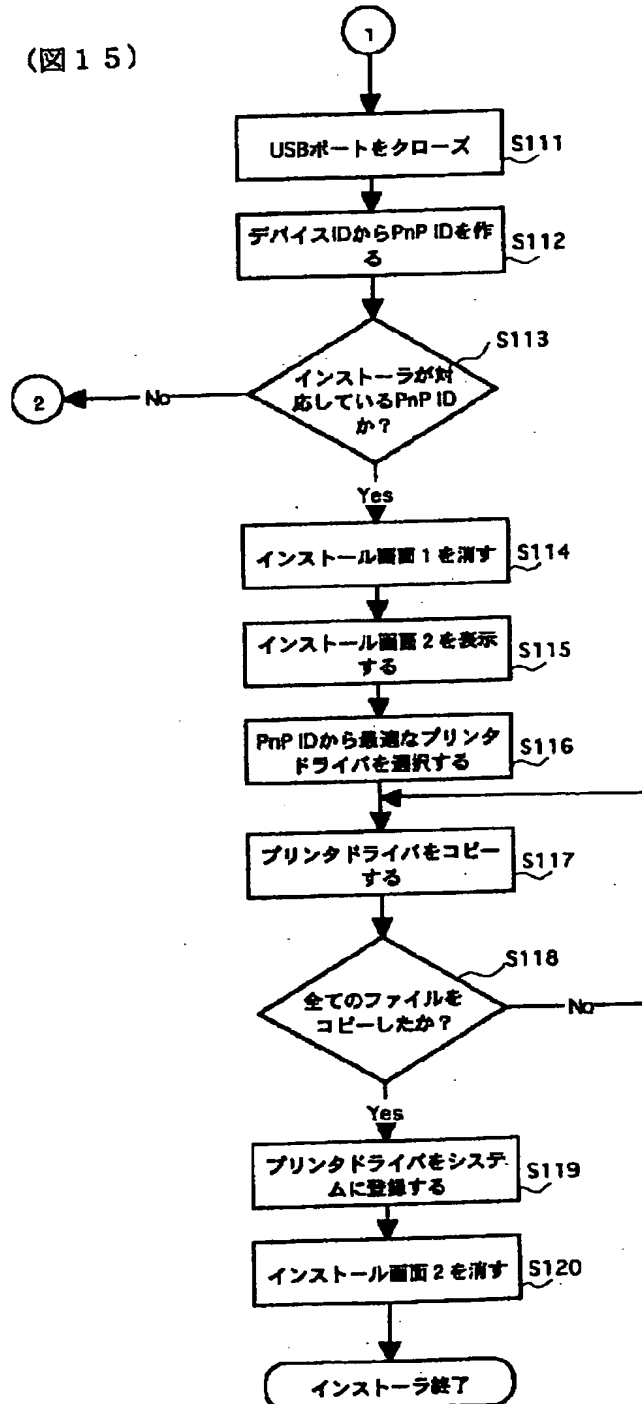
【図14】



実施形態2のフローチャート前半

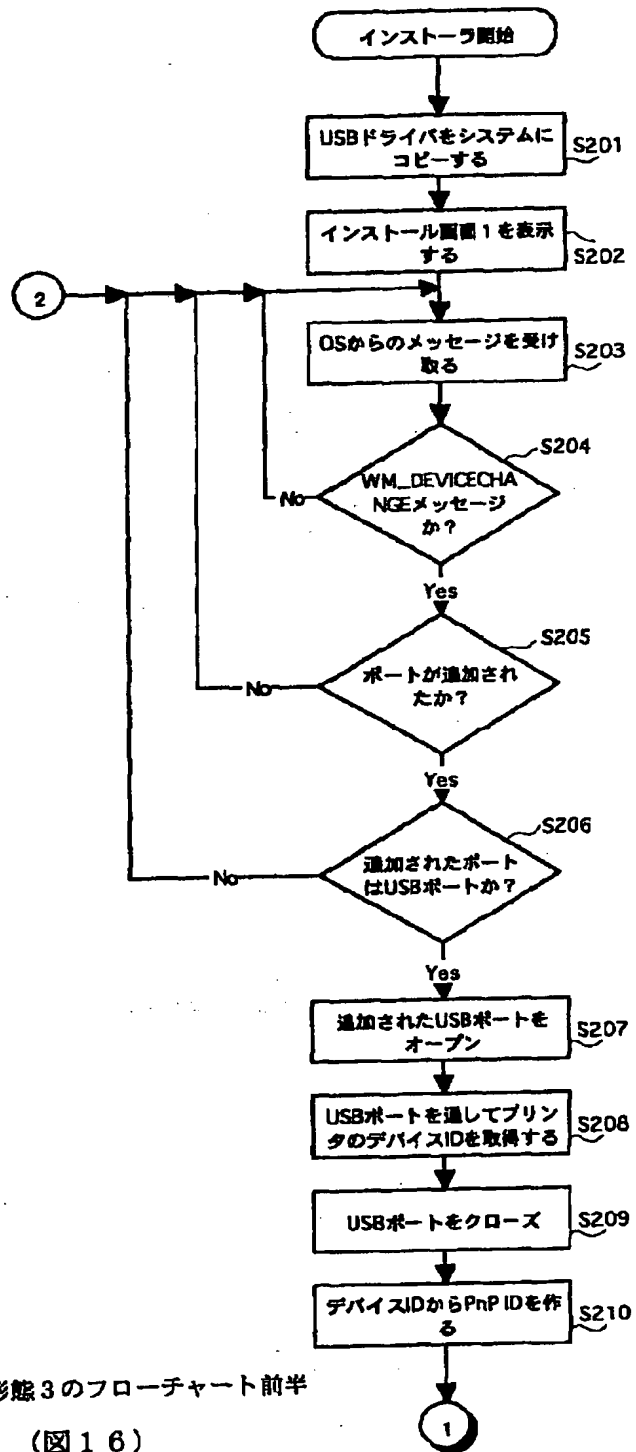
【図15】

(図15)



実施形態2のフローチャート後半

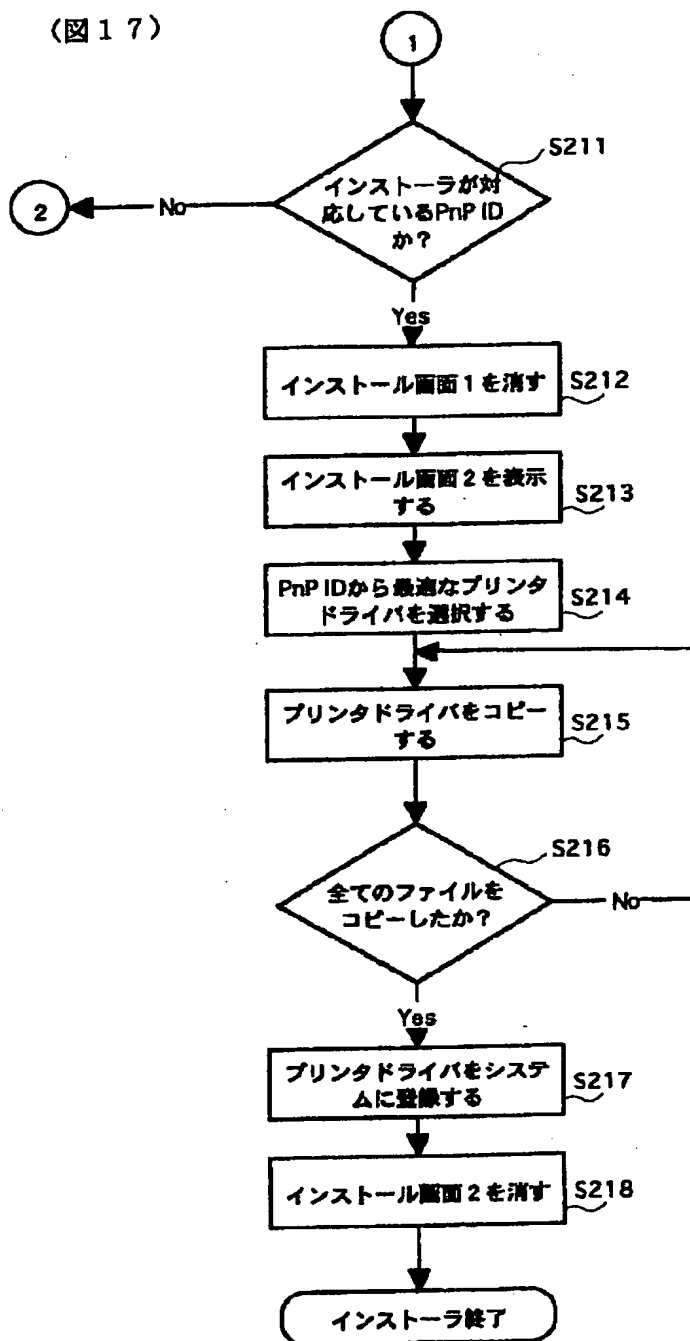
【図16】



実施形態3のフローチャート前半

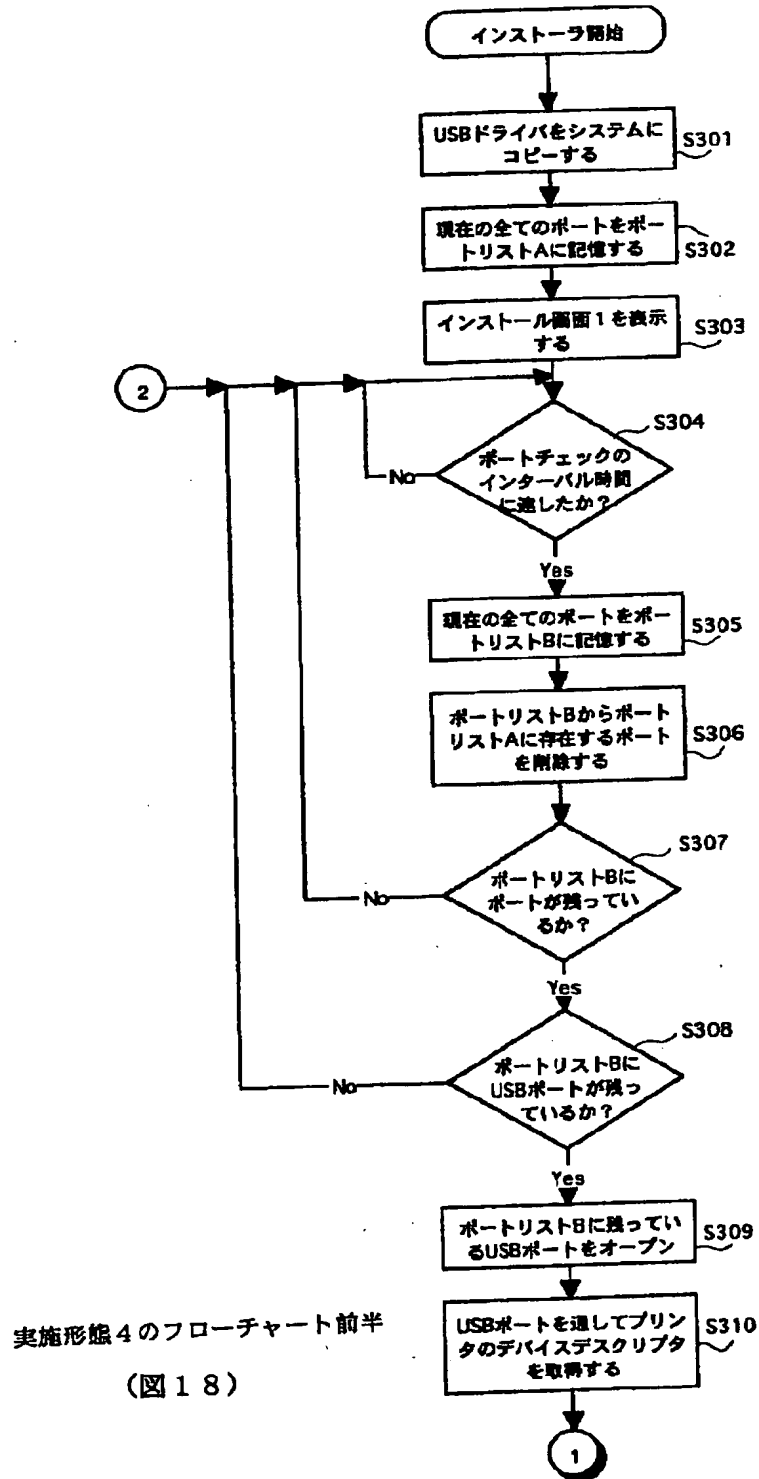
(図16)

【図17】



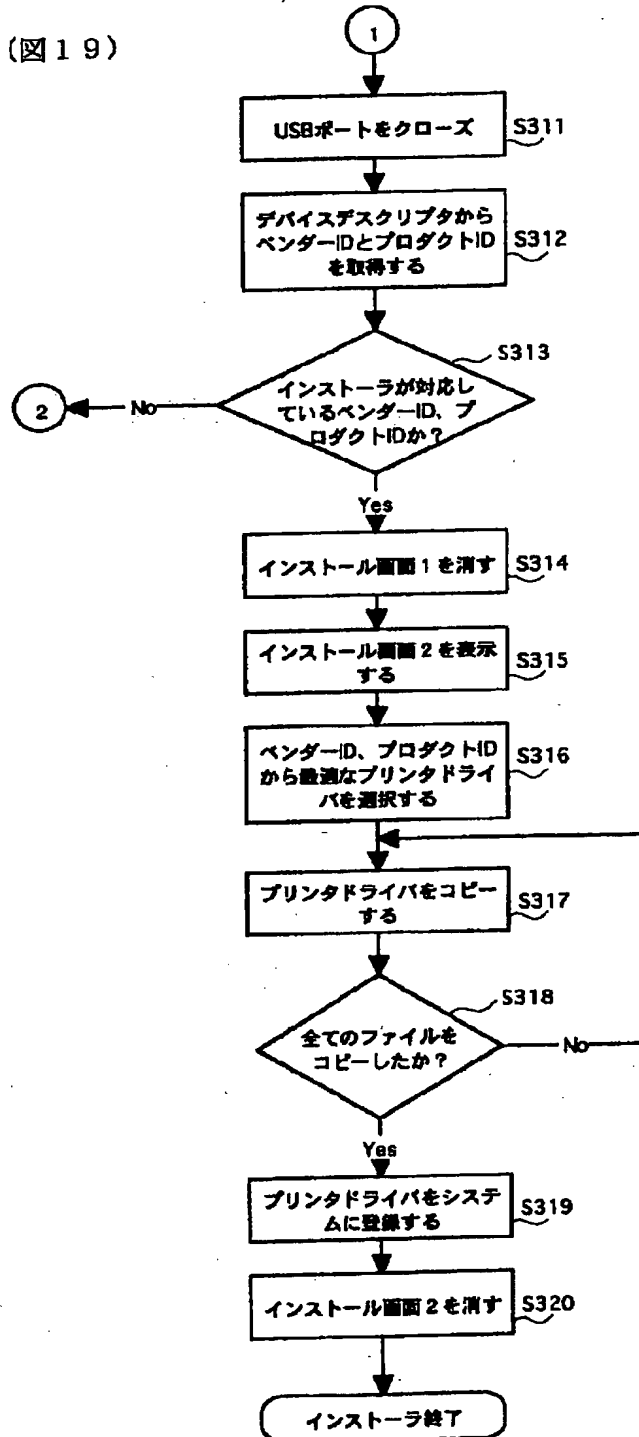
実施形態3のフローチャート後半

【図18】



【図19】

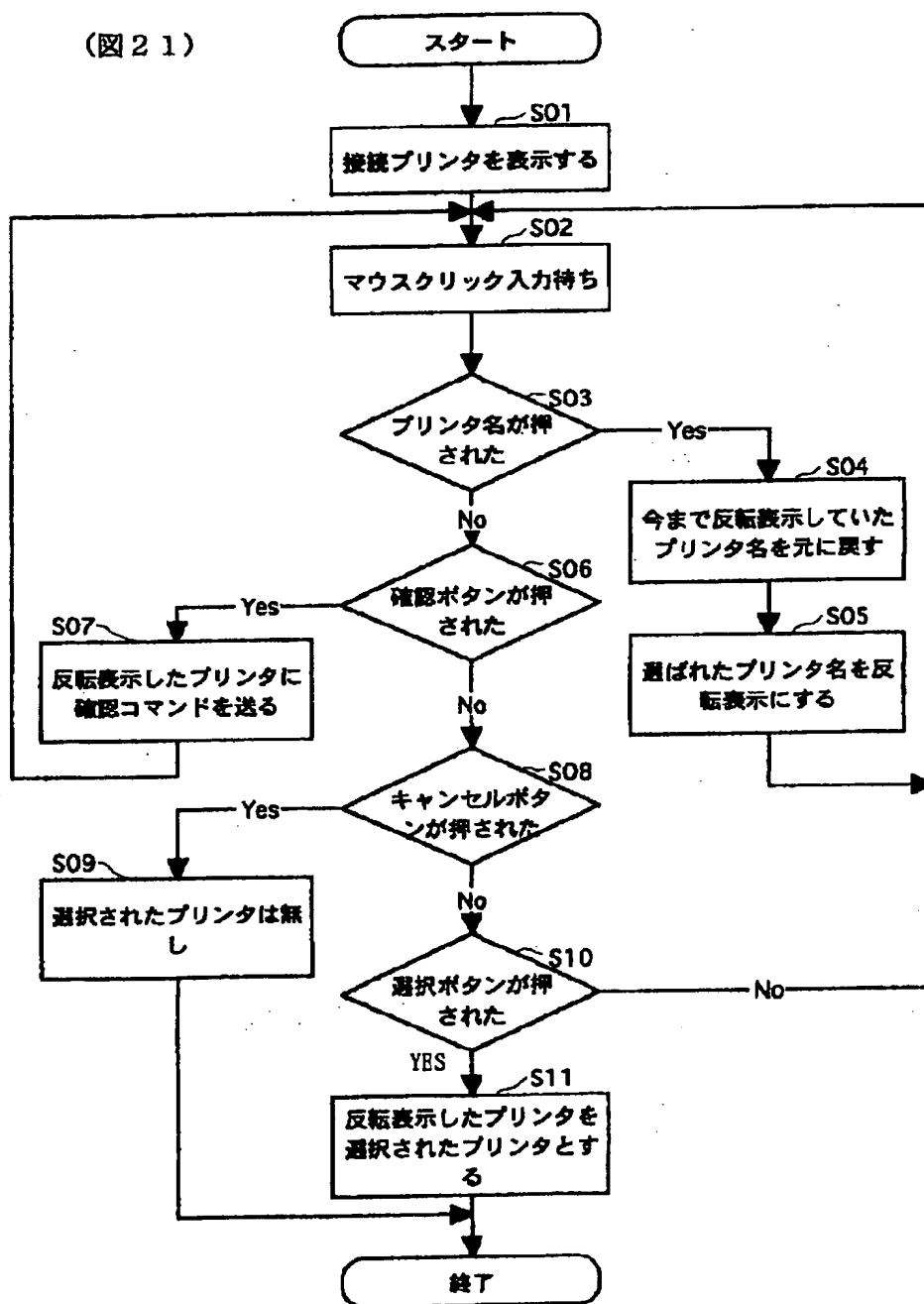
(図19)



実施形態4のフローチャート後半

【図21】

(図21)

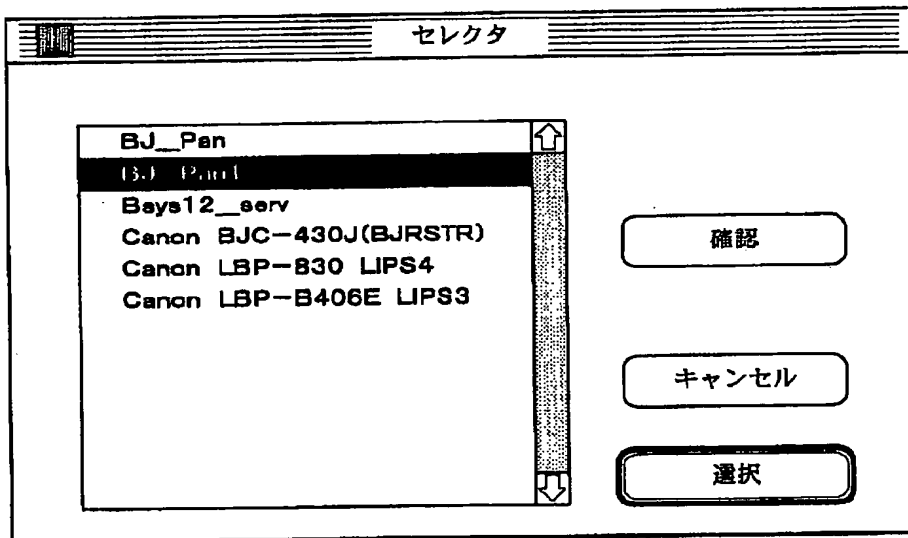


実施形態5のフローチャート



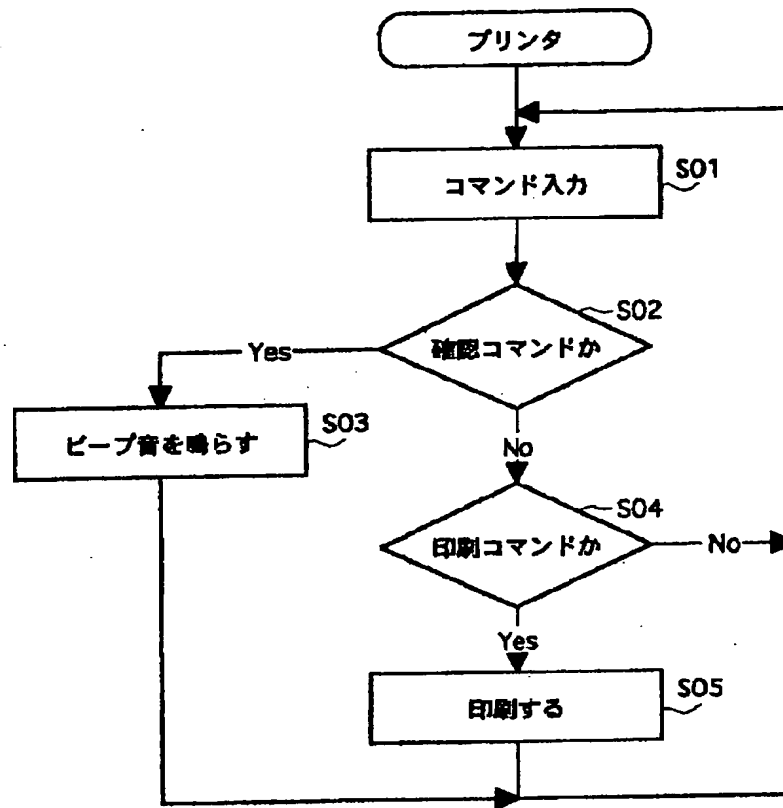
【図22】

(図22)



実施形態5のプリンタ選択ダイアログ

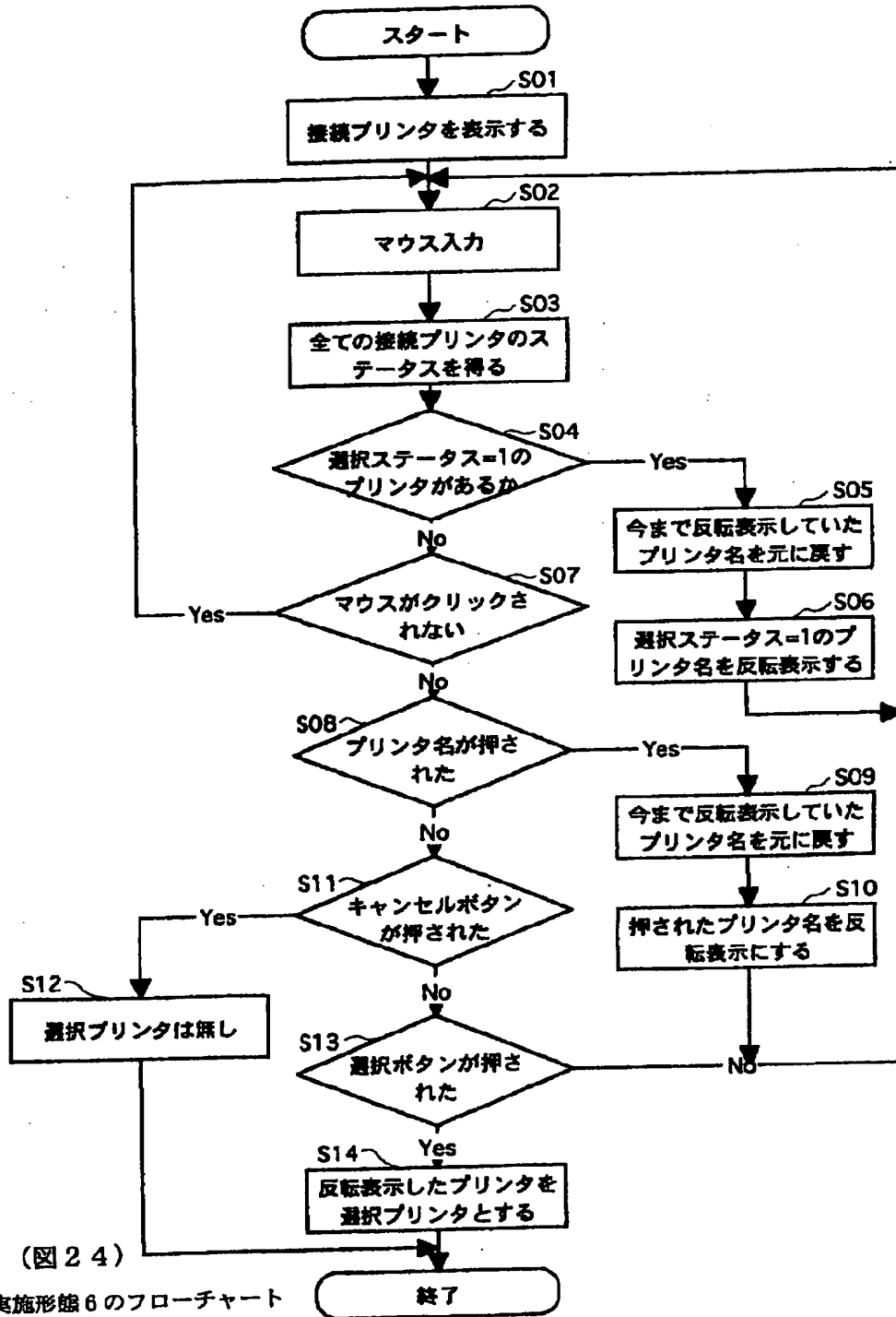
【図23】



プリンタのフローチャート

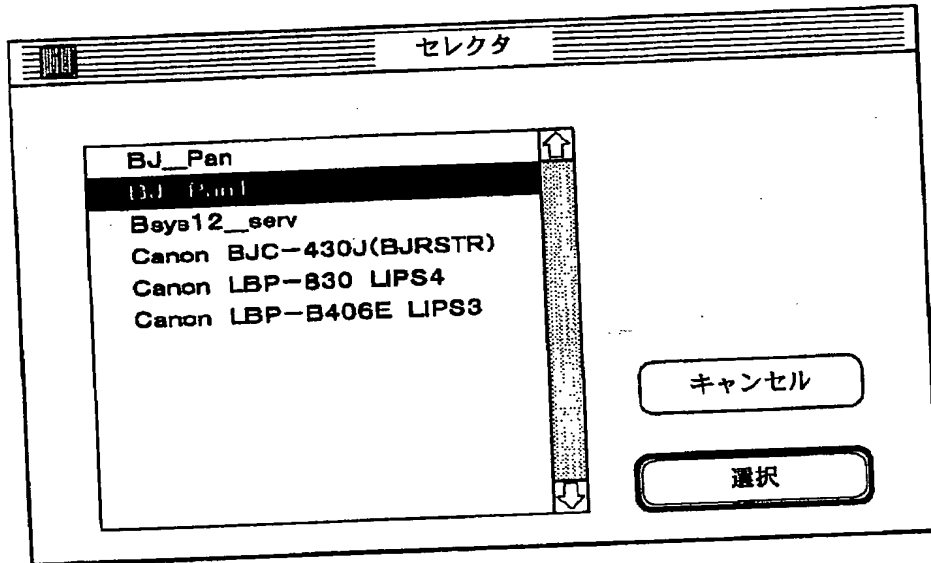
(図23)

【図24】

(図24)  
実施形態6のフローチャート

【図25】

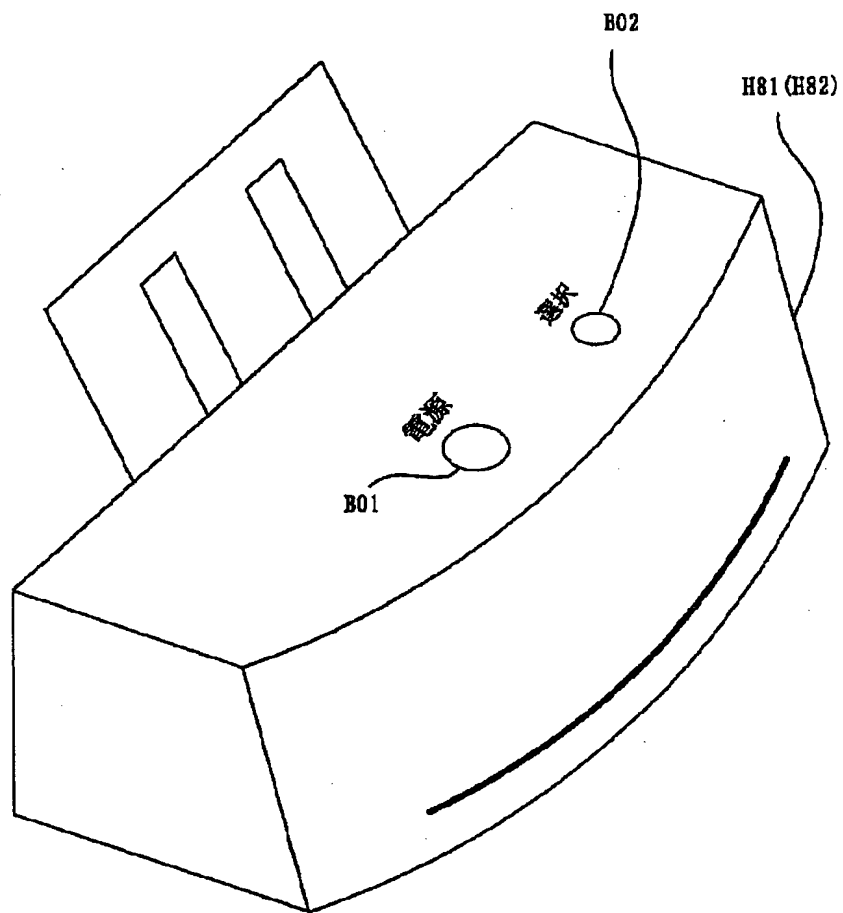
(図25)



実施形態6のプリンタ選択ダイアログ

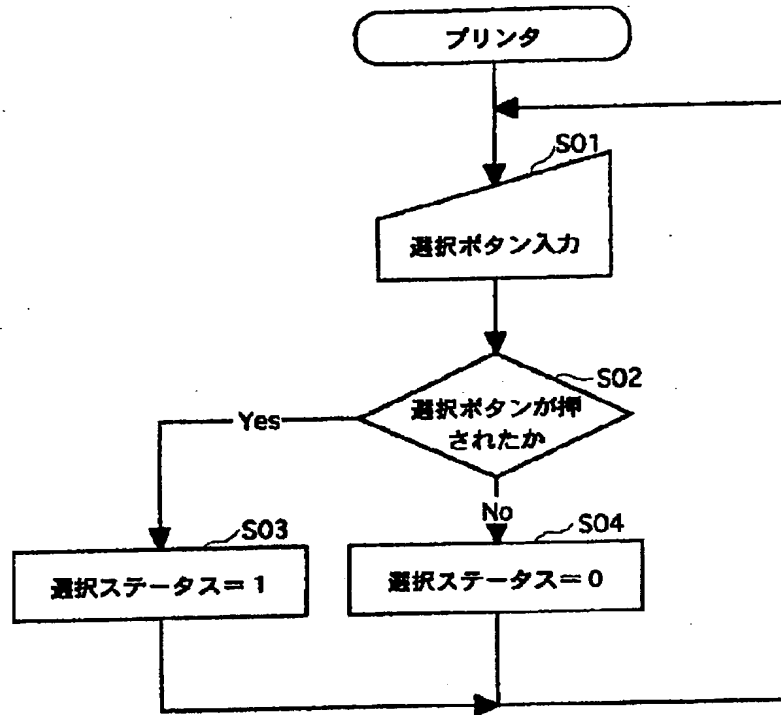
【図26】

(図26)



実施形態6の選択ボタン付きプリンタ

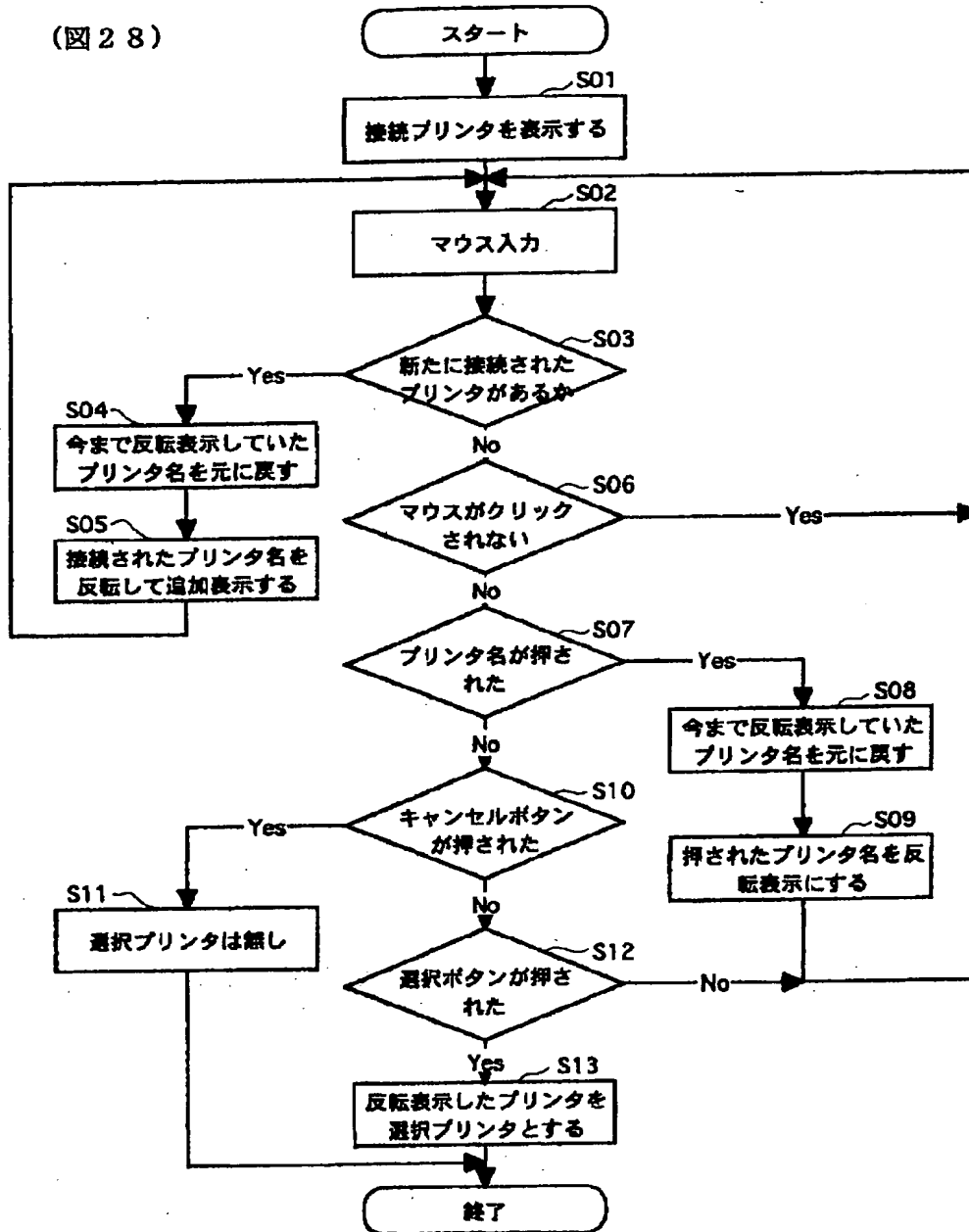
【図27】



選択ステータスのフローチャート

(図27)

【図28】



実施形態7のフローチャート